# CH32M030 Reference Manual

V1.1

https://wch-ic.com

## Overview

CH32M030 series is an industrial-grade motor MCU designed based on QingKe RISC-V3B core.CH32M030 has 4 built-in OPAs and 3 CMPs, which support combining into 2 groups of AC small-signal amplification decoder QII and 2 groups of differential input current sampling ISP; built-in USBFS PHY and PD PHY, supporting USB Host and USB Device function, PDUSB, Type-C fast charging function, BC1.2 and DCP/CDP and other high-voltage charging protocols; built-in pre-driver to provide high-voltage I/O; built-in programmable current-flooding module; provides a DMA controller, 12-bit ADC, multi-group timer, UART serial port, I2C, SPI, and other rich peripherals, and other resources. Provide over-voltage protection and over-temperature protection.

This manual provides detailed information on the use of CH32M030 series products for users' application development.

Please refer to the following datasheet for device characteristics of this family: *CH32M030DS0*.

**RISC-V core version comparison overview**

| Features / Core versions | Instruction set | Hardware stack levels | Interrupt nesting levels | Fast interrupt channels | Flow line | Vector table model | Extended instruction | Interface debug |
|---|---|---|---|---|---|---|---|---|
| QingKeV3B | IMCB | 2 | 2 | 4 | 3 | Address or command | Supported | None |

*For information about the RISC-V core, refer to the QingKeV3 Microprocessor Manual.*

**Abbreviated description of the bit attribute in the register:**

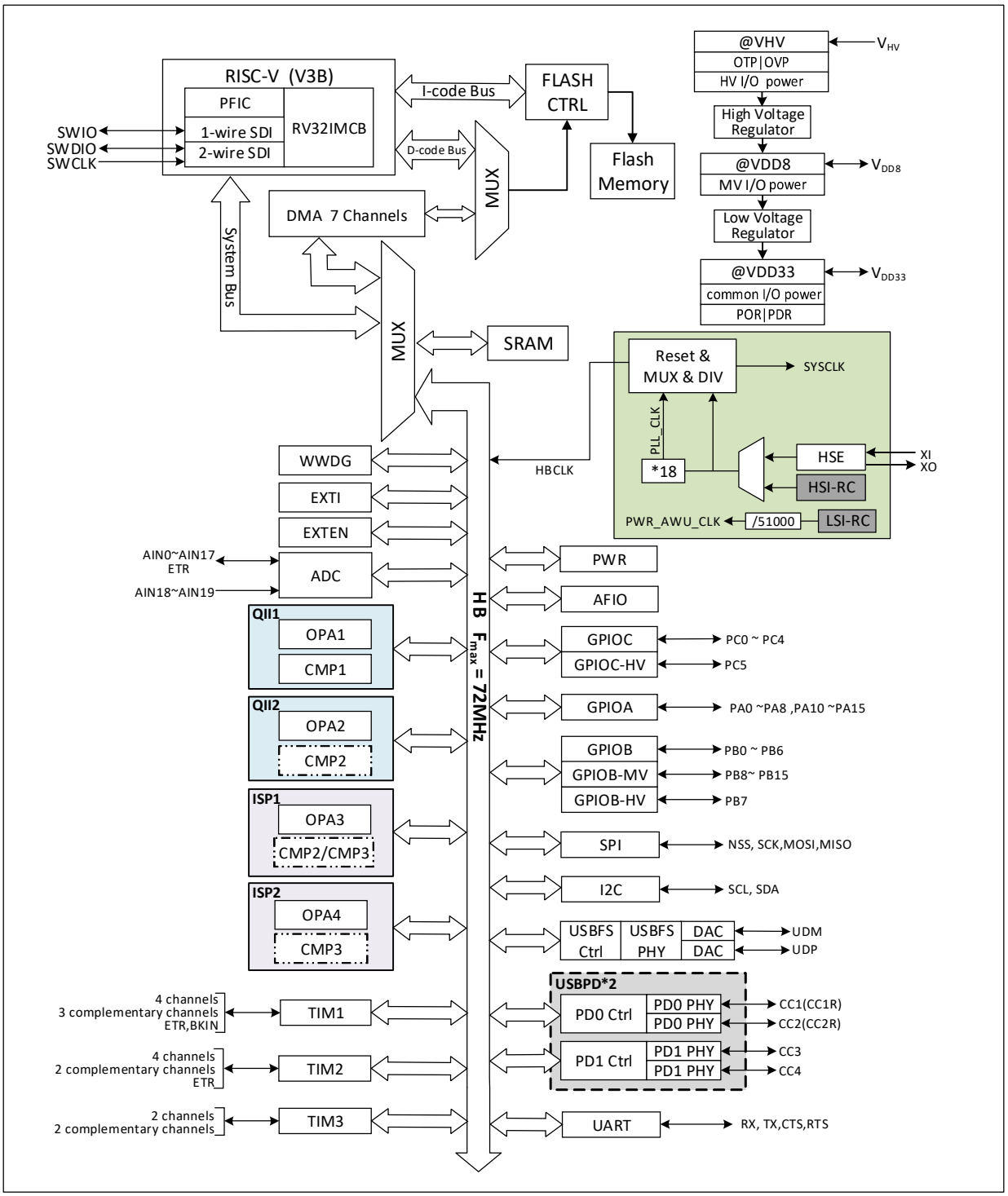
| Register bit attributes | Property description |
|---|---|
| RF | Read-only attribute, read a fixed value. |
| RO | Read-only attribute, changed by hardware. |
| RZ | Read-only attribute, auto bit clear 0 after read operation. |
| WO | Write only attribute (Not readable, read value uncertain) |
| WA | Write-only attribute, writable in Safe mode. |
| WZ | Write only attribute, auto bit clear 0 after write operation. |
| RW | Readable and writable. |
| RWA | Readable, writable in Safe mode. |
| RW1 | Readable, write 1 valid, write 0 invalid. |
| RW0 | Readable, write 0 valid, write 1 invalid. |

# Chapter 1 Memory and Bus Architecture

## 1.1 Bus Architecture

CH32M030 series products are based on RISC-V instruction set QingKe V3B design, its architecture will be the core, arbitration unit, DMA module, SRAM storage and other parts of the interaction through multiple buses. Designed to integrate a general-purpose DMA controller to reduce the burden on the CPU to improve access efficiency, while both data protection mechanisms, automatic clock switching protection and other measures to increase system stability. The system block diagram is shown in Figure 1-1.

Figure 1-1 CH32M030 system block diagram




The system is equipped with: Flash access prefetching mechanism to speed up code execution; General-purpose DMA controller to reduce the CPU burden and improve efficiency; clock tree hierarchy management to reduce the total power consumption of peripherals, as well as data protection mechanisms, clock security system protection mechanisms and other measures to increase system stability.

- The instruction bus (I-Code) connects the core to the FLASH instruction interface and prefetching is done on this bus.
- The data bus (D-Code) connects the core to the FLASH data interface for constant loading and debugging.

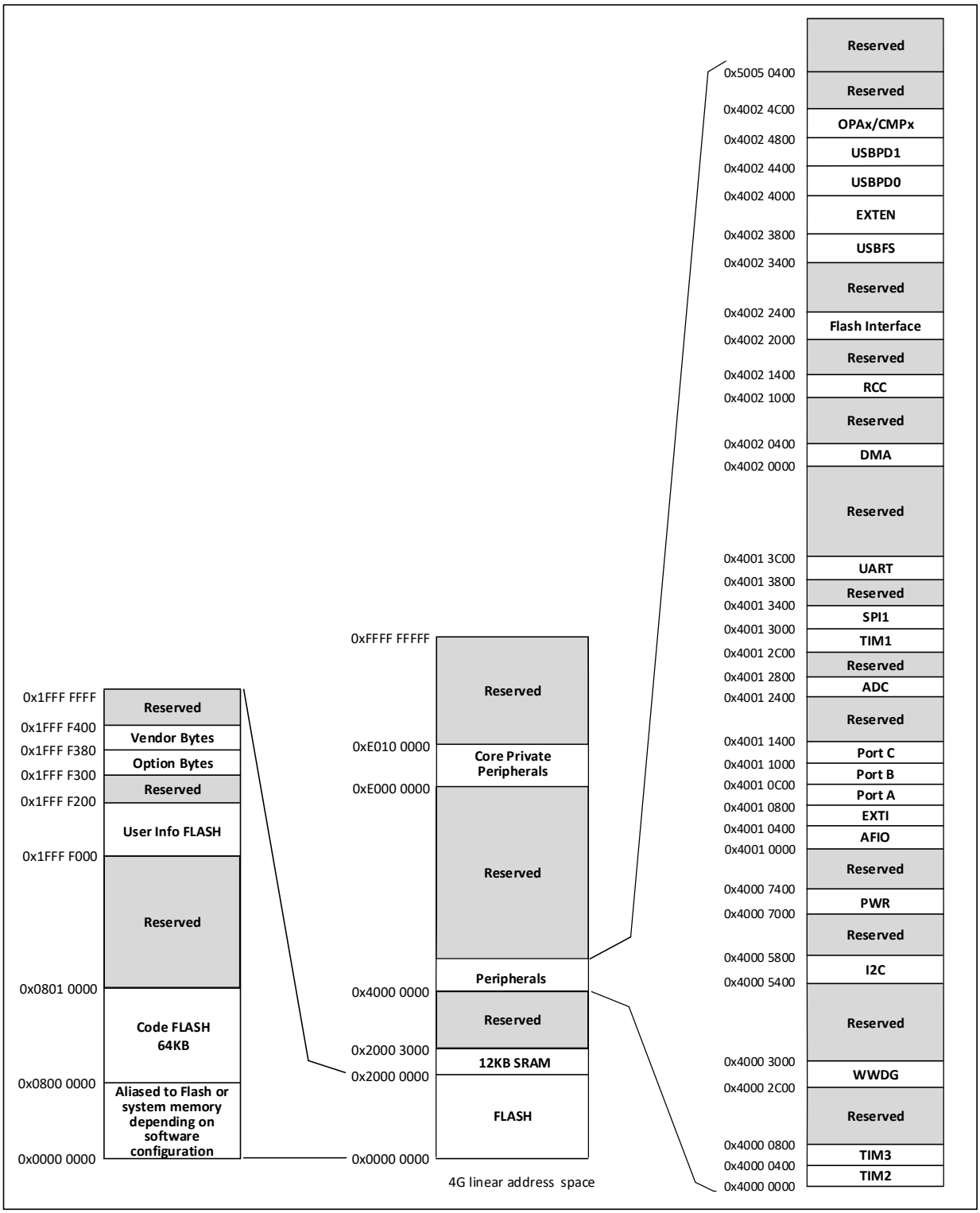
- The system bus connects the core to the bus matrix and is used to coordinate accesses to the core, DMA, SRAM and peripherals.
- The DMA bus is responsible for the DMA of the HB main control interface connected to the bus matrix, which is accessed by FLASH data, SRAM and peripherals.
- The bus matrix is responsible for the access coordination between the system bus, data bus, DMA bus, SRAM.

## 1.2 Memory Map

The CH32M030 family contains program memory, data memory, core registers, peripheral registers, and more, all addressed in a 4GB linear space.

System storage stores data in small-end format, i.e., low bytes are stored at the low address and high bytes are stored at the high address.

Figure 1-2 CH32M030 storage image



### 1.2.1 Memory Allocation

Built-in 12KB SRAM, starting address 0x20000000, supports byte, half-word (2 bytes), and full-word (4 bytes) access.

Built-in 64K bytes Code FLASH, i.e. user area, with ECC checksum, for user's application program and constant data storage.

Built-in 512-byte user-defined information storage area, operable only with the WCH-LinkUtility software tool.

Built-in 128-byte system non-volatile configuration information storage area for vendor's configuration word storage, factory-cured, not user-modifiable.

Built-in 128-byte user option byte storage area.

# Chapter 2 Power Control (PWR)

## 2.1 Overview

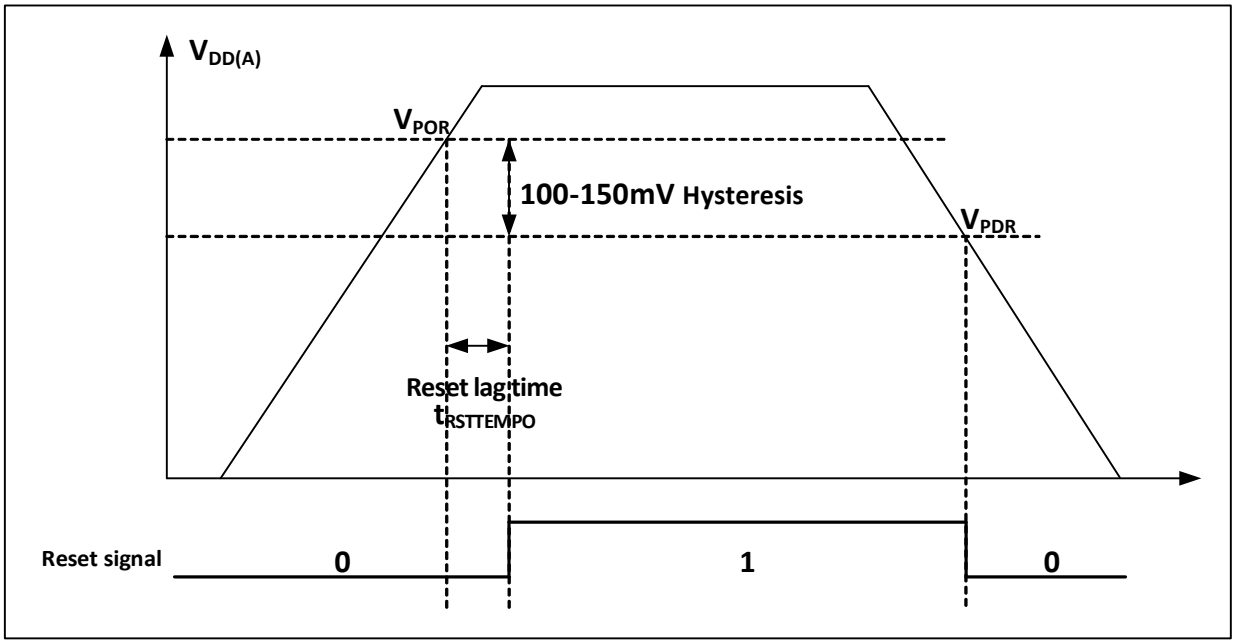
The system operating voltage $V_{HV}$ ranges 4.0~29.0V.

## 2.2 Power Management

### 2.2.1 Power-on Reset and Power-down Reset

The system integrated a power-on reset POR and a power-down reset PDR circuit. When the chip supply voltage $V_{DD}$ and $V_{DDA}$ falls below the corresponding threshold voltage, the system is reset by the relevant circuit, and no additional external reset circuit is required. Please refer to the corresponding datasheet for the parameters of the power-on threshold voltage $V_{POR}$ and the power-down threshold voltage $V_{PDR}$.

Figure 2-2 Schematic diagram of the operation of POR and PDR




### 2.2.2 Overvoltage Reset

By configuring the SEL_IO_VHV[1:0] bit, enable the PB4 port overvoltage protection detection. This function can automatically configure OVP overvoltage protection outside the chip through the PB4 pin. According to the threshold voltage of the internal judgment circuit and the proportion of the off-chip resistance voltage division, the overvoltage protection point of $V_{HV}$ can be set independently. When the $V_{HV}$ voltage is too high, the MCU can be forced to reset.

### 2.2.3 Divider Voltage Monitoring

Enable the PB4 port $V_{HV}$ divider voltage monitoring by configuring the SEL_IO_VHV[1:0] bits. This function can input $V_{HV}$ divider voltage through PB4 pin, and the voltage value will be sampled and monitored by ADC, and the real-time $V_{HV}$ value can be deduced according to the divider ratio of the off-chip resistor. For example, the external partial voltage resistor is 10K and 1K, at this time the ratio of sampling voltage and $V_{HV}$ is 1:11, if the ADC sampling

voltage value is 1V, then it can be deduced that the voltage value of $V_{HV}$ is 11V at this time.

## 2.3 Low-power Modes

After a system reset, the microcontroller is in a normal operating state (Run mode), where system power can be saved by reducing the system main frequency or turning off the unused peripheral clock or reducing the operating peripheral clock. If the system does not need to work, you can set the system to enter low-power mode and let the system jump out of this state by specific events.

Microcontrollers currently offer 3 low-power modes, divided in terms of operating differences between processors, peripherals, voltage regulators, etc.

- Sleep mode: The core stops running and all peripherals (Including core private peripherals) are still running.
- Stop mode: Stops all clocks and the system continues to run after waking up.
- Standby mode: Stops all clocks and the system continues to run after waking up.

Table 2-1 Low-power mode list

| Mode | Entry | Wake-up source | Effect on clock | Voltage regulator |
|---|---|---|---|---|
| Sleep | WFI | Any interrupt | Core clock OFF, no effect on other clocks | Standard mode |
| | WFE | Wake-up event | | |
| Stop | Set SLEEPDEEP to 1<br>Clear PDDS to 0<br>WFI or WFE | Any external interrupt/event | Disable HSE, HIS, PLL and peripheral clock | Standard mode |
| Standby | Set SLEEPDEEP to 1<br>Set PDDS to 1<br>WFI or WFE | Any external interrupt/event, RST pin reset | Disable HSE, HIS, PLL and peripheral clock | Low-power mode |

### 2.3.1 Low-power Configuration Options

- WFI and WFE

WFI: The microcontroller is woken up by an interrupt source with interrupt controller response, and the interrupt service function will be executed first after the system wakes up (Except for microcontroller reset).

WFE: The wakeup event triggers the microcontroller to exit low-power mode. Wake-up events include.

1) Configure an external or internal EXTI line to event mode, when no interrupt controller needs to be configured.
2) Or configure an interrupt source, equivalent to a WFI wakeup, where the system prioritizes the execution of the interrupt service function.
3) Or configure the SEVONPEND bit to turn on peripheral interrupt enable, but not interrupt enable in the interrupt controller, and the interrupt pending bit needs to be cleared after the system wakes up.

- SLEEPONEXIT

  Enable: After executing the WFI or WFE instruction, the microcontroller ensures that all pending interrupt services are exited and then enters low-power mode.

  Not enabled: The microcontroller enters low-power mode immediately after executing the WFI or WFE command.

- SEVONPEND

  Enable: All interrupts or wake-up events can wake up the low-power consumption entered by executing WFE.

Not enabled: Only interrupts or wake-up events enabled in the interrupt controller can wake up the low-power consumption entered by executing WFE.

### 2.3.2 Sleep Mode

In this mode, all I/O pins keep their state in Run mode and all peripheral clocks are normal, so try to turn off useless peripheral clocks before entering Sleep mode to reduce low-power consumption. This mode takes the shortest time to wake up.

Enter: Configure core register control bit SLEEPDEEP=0, power control register PDDS=0, execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exit: Arbitrary interrupt or wakeup event.

### 2.3.3 Stop Mode

Stop mode is based on the deep sleep mode (SLEEPDEEP) of the core combined with peripheral clock control mechanism. This mode high frequency clock (HSE/HSI/PLL) domain is turned off, SRAM and register contents are maintained, and IO pin state is maintained. In this mode the high frequency clock (HSI) domain is switched off, the SRAM and register contents are maintained and the IO pin state is held. The system can continue to run after this mode wakes up and the HSI is called the default system clock.

If flash programming is in progress, the system does not enter stop mode until access to memory is complete. If access to the HB is being carried out, the system does not enter stop mode until the access to the HB is completed.

Operating module in Stop mode: Low frequency clock (LSI).

Enter: Configure the core register control bit SLEEPDEEP=1, PDDS=0 in the power control register, execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exits: Any external interrupt/event.

### 2.3.4 Standby Mode

The only difference between Standby mode and Stop mode is that in standby mode, the internal voltage regulator will enter low-power mode.

Operating module in Standby mode: Low frequency clock (LSI).

Enter: Configure the core register control bit SLEEPDEEP=1, PDDS=1 in the power control register, and execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exit:

1) Any external interrupt/event.
2) External reset on RST pin.

*Note: In debug mode, if the microprocessor enters Stop or Standby mode, the debug connection will be lost.*

### 2.3.5 Auto-wakeup (AWU)

The AWU module enables automatic wake-up without external interrupts. Wake-up from stop or standby mode can be done periodically by programming the time base.

When the AWU interrupt function is enabled, the rising/falling edge trigger of the 17th line connected to the EXTI module needs to be set to enable this interrupt (configure EXTI).

Internal low-frequency clock oscillator LSI (340kHz) divides 51000 as automatic wake-up counting time base.

## 2.4 Register Description

Table 2-2 PWR-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_PWR_CTLR | 0x40007000 | Power control register | 0x00000400 |
| R32_PWR_AWUCSR | 0x40007008 | AWU control/status register | 0x00000000 |
| R32_PWR_AWUWR | 0x4000700C | AWU window comparison value register | 0x00000000 |

### 2.4.1 Power Control Register (PWR_CTLR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | PMUPROEN | SEL_IO_VHV[1:0] | | Reserved | ISINKEN | Reserved | | | | | | | | PDDS | Reserved |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved | 0 |
| 14 | PMUPROEN | RW | PMU over-temperature protection enable:<br>1: Enable over-temperature protection;<br>0: Disable over-temperature protection. | 1 |
| [13:12] | SEL_IO_VHV[1:0] | RW | PB4 port function selection bit:<br>00: Floating;<br>01: GPIO function;<br>10: $V_{HV}$ voltage divider monitoring;<br>11: Overvoltage protection detection.<br>*Note: (1) If this bit is operated, the reset value will be the default value only after power-on reset, and the reset values of other reset methods will remain the original write values;*<br>*(2) Before configuring this bit, you need to configure the user-selected word OV_CFG bit.* | 0 |
| 11 | Reserved | RO | Reserved | 0 |
| 10 | ISINKEN | RW | ISINK enable:<br>1: Enable;<br>0: Disable. | 0 |
| [9:2] | Reserved | RO | Reserved | 0 |
| 1 | PDDS | RW | In deep sleep scenario, standby/stop mode selection bit:<br>1: Enter Standby mode;<br>0: Enter Stop mode. | 0 |
| 0 | Reserved | RO | Reserved | 0 |

### 2.4.2 AWU Control/Status Register (PWR_AWUCSR)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | AWUEN | Reserved |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:2] | Reserved | RO | Reserved | 0 |
| 1 | AWUEN | RW | AWU enable:<br>1: Enable AWU;<br>0: Invalid. | 0 |
| 0 | Reserved | RO | Reserved | 0 |

### 2.4.3 AWU Window Comparison Value Register (PWR_AWUCSR)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | AWUWR[3:0] | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:6] | Reserved | RO | Reserved | 0 |
| [3:0] | AWUWR[3:0] | RW | AWU window value:<br>The AWU window value is equal to the input value of the AWU window value + 1;<br>The AWU window value is used to compare with the incremental counter value and generate a wake-up signal when the counter value is equal to the window value. | 0 |

# Chapter 3 Reset and Clock Control (RCC)

The controller provides different forms of resets and configurable clock tree structures based on the division of power areas and peripheral power management considerations in the application. This section describes the scope of each clock in the system.

## 3.1 Main Features

- Multiple reset forms
- Multiple clock sources, bus clock management
- Independent management of each peripheral clock: Reset, On, Off
- Supports internal clock output

## 3.2 Reset

The controller provides 4 forms of reset: Power Reset, System Reset, Overtemperature Reset and Overvoltage Reset.

### 3.2.1 Power Reset

When a power Reset occurs, it will reset all registers.

A power Reset is generated when the following event occurs:

- Power-up/power-down reset (POR/PDR), triggered when the $V_{HV}$ voltage drops to make the $V_{DD33}$ voltage lower than the set threshold ($V_{POR/PDR}$).

### 3.2.2 System Reset

When a system Reset occurs, it will reset the reset flag in addition to the control/status register RCC_RSTSCKR and all the registers. The source of the reset event is identified by looking at the reset status flag bit in the RCC_RSTSCKR register.

A system Reset is generated when one of the following events occurs:

- Low signal on RST pin (External reset)
- Window watchdog count termination (WWDG reset)
- Software reset (SW reset)
- USBPD reset
- Low-power management reset
- ADC reset
- OVP overvoltage reset

Window/Independent Watchdog Reset: Generated by the window/independent watchdog peripheral timer count cycle overflow trigger, see its corresponding section for detailed description.

Software reset: CH32M030 product resets the system through the SYSRESET bit 1 of the interrupt configuration register PFIC_SCTLR in the programmable interrupt controller PFIC. Please refer to the corresponding chapter for details.

USBPD reset: When PD_RST_EN is 1, CH32M030 supports the reset generated by USB PD signal frame Hard Reset; If IE_RX_RESET is also 1, it also supports the reset generated by the signal frame Cable Reset. USB PD has

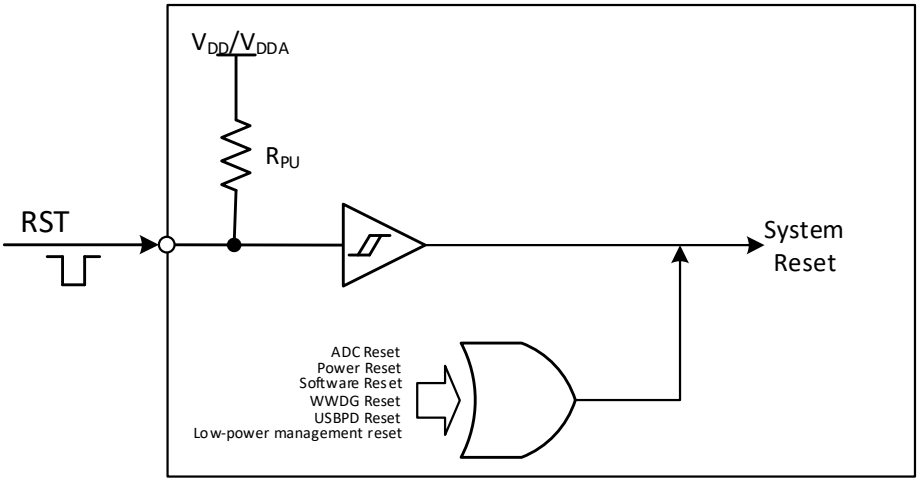
no reset flag, but it produces the same reset effect as software reset.

Low Power Management Reset: Standby mode reset is enabled by setting the STANDYRST bit in the user option byte. At this time, after entering the Standby mode, the system reset will be performed instead of entering the standby mode. Stop mode reset is enabled by setting the STOP_RST bit in the user option byte. At this time, after the process of entering the Stop mode is performed, the system reset will be performed instead of entering the Stop mode.

ADC Reset: If the ADC watchdog reset enable is on, an ADC reset is generated when the ADC data is greater than the watchdog high threshold or less than the watchdog low threshold.

OVP overvoltage reset: Triggered when the $V_{HV}$ voltage is higher than the set threshold voltage.

Figure 3-1 System reset structure



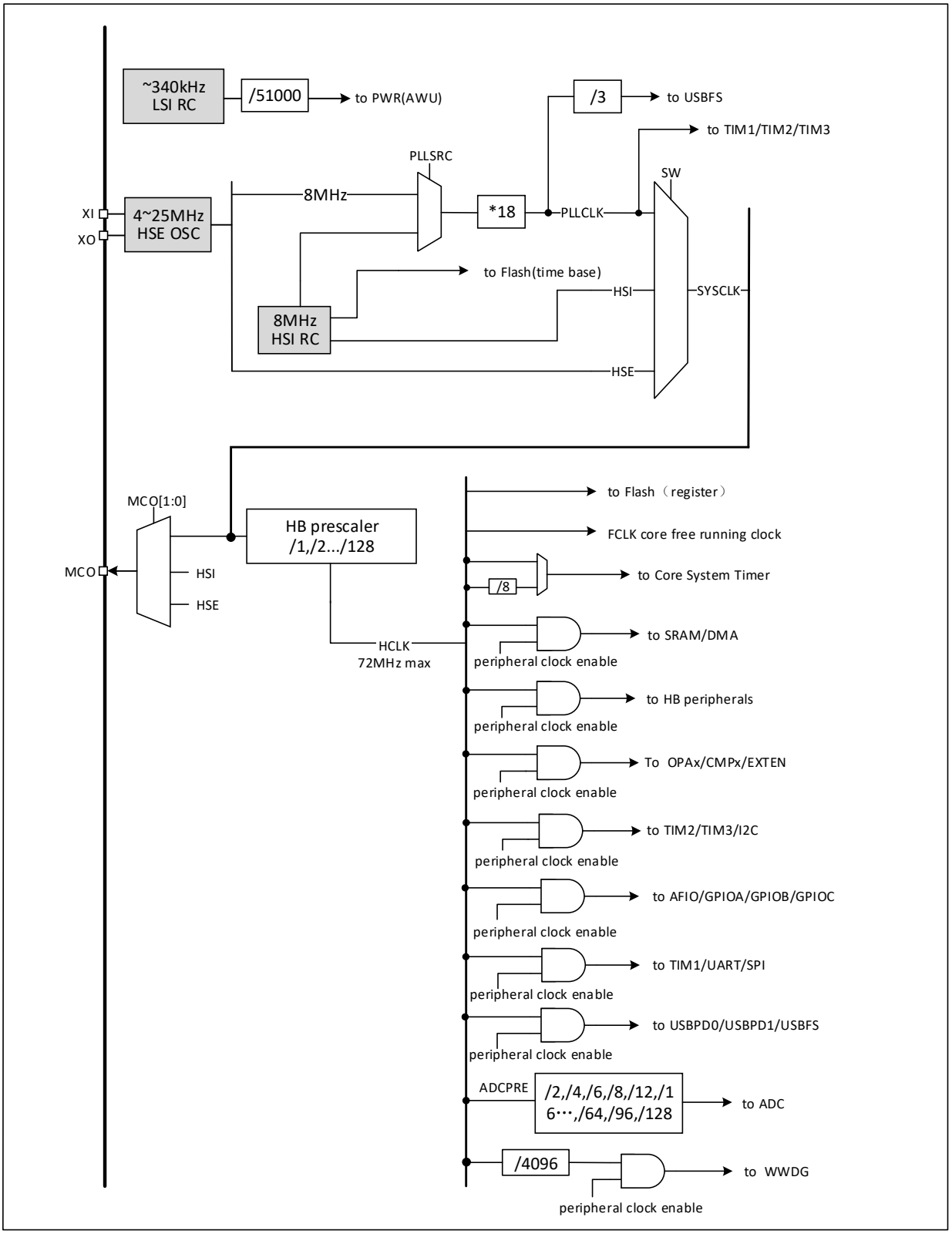

### 3.2.3 Overtemperature Reset

CH32M030 has built-in OTP over-temperature protection module. By setting PMUPROEN bit to 1, PMU over-temperature protection is enabled, which will forcibly reset MCU when the chip temperature is too high.

# 3.3 Clock

## 3.3.1 System Clock Structure

Figure 3-2 Clock tree block diagram



### 3.3.2 High-speed Clock (HSI)

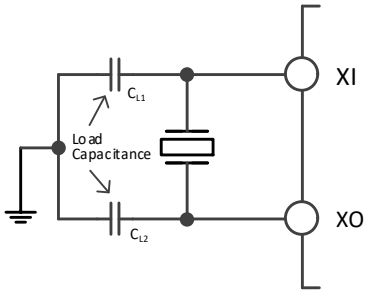
HSI is a high-speed clock signal generated by the system internal 8MHz RC oscillator. HSI RC oscillator can provide system clock without any external devices. It has a short start-up time. HSI is enabled and disabled by setting the HSION bit in the RCC_CTLR register, and the HSIRDY bit indicates whether the HSI RC oscillator is stable or not. The system defaults HSION and HSIRDY to 1 (It is recommended not to turn them off). If the HSIRDYIE bit in the RCC_INTR register is set, a corresponding interrupt will be generated.

- Factory calibration: The difference of manufacturing process will cause different RC oscillation frequency for each chip, so HSI calibration is performed for each chip before it is shipped. After system reset, the factory calibration value is loaded into HSICAL[7:0] of the RCC_CTLR register.
- User tuning: Based on different voltages or ambient temperatures, the application can adjust the HSI frequency by using the HSITRIM[4:0] bits in the RCC_CTLR register.

HSE is an external high-speed clock signal, including external crystal/ceramic resonator generation or external high-speed clock input.

- External crystal/ceramic resonator (HSE crystal): An external oscillator provides a more accurate clock source for the system. Further information can be found in the electrical characteristics section of the data sheet. The HSE crystal can be turned on and off by setting the HSEON bit in the RCC_CTLR register, and the HSERDY bit indicates whether the oscillation of the HSE crystal is stable, and the hardware sends the clock to the system only after the HSERDY bit is set. If the HSERDYIE bit in the RCC_INTR register is set, an interrupt will be generated.

Figure 3-3 High speed external crystal circuit




*Note: The load capacitance should be as close as possible to the oscillator pin, and the capacitance value should be selected according to the crystal manufacturer's parameters.*

- External high-speed clock source (HSE bypass): In this mode, the clock source is directly sent to the XI pin from the outside, and the XO pin is suspended. The application program needs to set the HSEBYP bit and turn on the HSE bypass function when the HSEON bit is 0, and then set the HSEON bit.

- Figure 3-4 High speed clock source circuit

### 3.3.3 Low-speed Clock (LSI)

LSI is a low-speed clock signal generated by RC oscillator inside the system. It can keep running in standby mode and provide a clock reference for the wake-up unit. Further information can be found in the electrical characteristics section of the data sheet. The LSI can be turned on and off by setting the LSION bit in the RCC_RSTSCKR register, and then checking whether the LSIRC oscillation is stable by querying the LSIRDY bit. The hardware sends the clock after the LSIRDY bit is set. If the LSIRDYIE bit in the RCC_INTR register is set, an interrupt will be generated.

### 3.3.4 PLL Clock (PLLCLK)

By configuring the PLLSRC bit in RCC_CFGR0 register, 2 clock sources can be selected for the internal PLL clock. These settings must be completed before the PLL is turned on. Once the PLL is turned on, these parameters cannot be changed. Set the PLLON bit in RCC_CTLR register to be turned on and off, and the PLLRDY bit indicates whether the PLL clock is stable, and the hardware sends the clock to the system after the PLL bit is set to 1. If the PLLRDYIE bit in the RCC_INTR register is set, the corresponding interrupt will be generated.

PLL clock source:

- HSI uncrossover feed
- HSE uncrossover feed

### 3.3.5 Bus/Peripheral Clock

#### 3.3.5.1 System Clock (SYSCLK)

Configure the system clock source by configuring bits SW[1:0] in RCC_CFGR0 register, and SWS[1:0] indicates the current system clock source.

- HSI as system clock
- HSE as system clock
- PLL clock as system clock

After the controller is reset, the default HSI clock is selected as the system clock source. Switching between clock sources will not happen until the target clock source is ready.

#### 3.3.5.2 HB Bus Peripheral Clock (HCLK)

The HB bus clocks can be configured by configuring the PPRE[6:0] bits of the RCC_CFGR0 register. The bus clock determines the peripheral interface access clock reference that is mounted below them. Applications can adjust different values to reduce the power consumption when some of the peripherals are operating.

The various bits in the RCC_AHBRSTR, RCC_APB1PRSTR and RCC_APB2PRSTR registers can reset the different peripheral modules to their initial state.

Each bit in the RCC_AHBPCENR, RCC_APB1PCENR, and RCC_APB2PCENR registers can be used to individually turn on or off the communication clock interface for different peripheral modules. When using a peripheral, you first need to turn on its clock enable bit in order to access its registers.

#### 3.3.5.3 Microcontroller Clock Output (MCO)

The microcontroller allows the clock signal to be output to the MCO pin. Configure the alternate push-pull output

mode in the corresponding GPIO port register, and set the MCO[2:0] bits in RCC_CFGR0 register to select the following clock signals as the MCO clock output:

- System clock (SYSCLK) output
- HSI clock output
- HSE clock output

## 3.4 Register Description

Table 3-1 RCC-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_RCC_CTLR | 0x40021000 | Clock control register | 0x0000xx83 |
| R32_RCC_CFGR0 | 0x40021004 | Clock configuration register0 | 0x00000000 |
| R32_RCC_INTR | 0x40021008 | Clock interrupt register | 0x00000000 |
| R32_RCC_PB2PRSTR | 0x4002100C | PB2 peripheral reset register | 0x00000000 |
| R32_RCC_PB1PRSTR | 0x40021010 | PB1 peripheral reset register | 0x00000000 |
| R32_RCC_HBPCENR | 0x40021014 | HB peripheral clock enable register | 0x00000004 |
| R32_RCC_PB2PCENR | 0x40021018 | PB2 peripheral clock enable register | 0x00000000 |
| R32_RCC_PB1PCENR | 0x4002101C | PB1 peripheral clock enable register | 0x00000000 |
| R32_RCC_RSTSCKR | 0x40021024 | Control/status register | 0x08000000 |
| R32_RCC_HBPRSTR | 0x40021028 | HB peripheral reset register | 0x00000000 |

### 3.4.1 Clock Control Register (RCC_CTLR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | PLLRDY | PLLON | Reserved | | | | | HSEBYP | HSERDY | HSEON |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSICAL[7:0] | | | | | | | | HSITRIM[4:0] | | | | | Reserved | HSIRDY | HSION |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:26] | Reserved | RO | Reserved | 0 |
| 25 | PLLRDY | RO | PLL clock ready lock flag bit (set by hardware):<br>1: PLL clock lock;<br>0: PLL clock is not locked. | 0 |
| 24 | PLLON | RW | PLL clock enable control bits:<br>1: Enable PLL clock;<br>0: Disable the PLL clock.<br>*Note: This bit is cleared by hardware after entering the stop or standby low power mode.* | 0 |
| [23:19] | Reserved | RO | Reserved | 0 |
| 18 | HSEBYP | RW | External high-speed crystal bypass control bits:<br>1. Bypass external high-speed crystal/ceramic resonator (using external clock source);<br>2. 0: High-speed external crystal/ceramic resonator is not bypassed.<br>*Note: This bit needs to be written when HSEON is 0.* | 0 |
| 17 | HSERDY | RO | External high-speed crystal oscillation stability ready flag (set by hardware):<br>1: External high-speed crystal oscillation stability; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0: External high-speed crystal oscillation is not stable.<br>*Note: After the HSEON bit is cleared, it takes 6 HSE cycles to clear it.* | |
| 16 | HSEON | RW | External high-speed crystal oscillation enabling control bits:<br>1. Enable HSE oscillator;<br>0: Disable HSE oscillator.<br>*Note: This bit is cleared by hardware after entering the stop or standby low power mode.* | 0 |
| [15:8] | HSICAL[7:0] | RO | The internal high-speed clock calibration value is automatically initialized when the system is started. | x |
| [7:3] | HSITRIM[4:0] | RW | Internal high-speed clock adjustment value:<br>Users can input an adjustment value and superimpose it on the value of HSICAL[7:0] to adjust the frequency of internal HSIRC oscillator according to the change of voltage and temperature.<br>The default value is 16, and the HSI can be adjusted to 8 MHz 0.25%; The change of HSICAL in each step is adjusted by about 22kHz. | 10000b |
| 2 | Reserved | RO | Reserved | 0 |
| 1 | HSIRDY | RO | Internal high-speed clock (8MHz) stable ready flag (set by hardware):<br>1: Internal high-speed clock (8MHz) stable;<br>0: Internal high-speed clock (8MHz) is not stable.<br>*Note: It takes 6 HSI cycles to clear the HSION bit after it is cleared.* | 1 |
| 0 | HSION | RW | Internal high-speed clock (8MHz) enable control bits:<br>1: Enable HSI oscillator;<br>0: Disable HSI oscillator.<br>*Note: This bit is set by hardware to 1 to start the internal 8MHz RC oscillator when returning from standby and stop modes or when the external oscillator HSE, which is used as the system clock, fails.* | 1 |

### 3.4.2 Clock Configuration Register0 (RCC_CFGR0)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | ADC_DUTY_SEL | Reserved | MCO[2:0] | | | Reserved | | | | | | | PLLSRC |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADCPRE[4:0] | | | | | HPRE[6:0] | | | | | | | SWS[1:0] | | SW[1:0] | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:29] | Reserved | RO | Reserved | 0 |
| 28 | ADC_DUTY_SEL | RW | ADC clock duty cycle selection:<br>1: ADC clock duty cycle is 75%;<br>0: ADC clock duty cycle is 50%. | 0 |
| 27 | Reserved | RO | Reserved | 0 |
| [26:24] | MCO[2:0] | RW | Clock output control of microcontroller MCO pin:<br>0xx: No clock output; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 100: System clock (SYSCLK) output;<br>101: Internal 8MHz RC oscillator clock (HSI) output;<br>110: External oscillator clock (HSE) output;<br>111: Reserved.<br>*Note: When the MCO clock is started or switched, several cycles of the clock may be lost. Ensure that the output clock frequency does not exceed 30mhz (the highest frequency of I/O port).* | |
| [23:17] | Reserved | RO | Reserved | 0 |
| 16 | PLLSRC | RW | Input clock source of PLL (write only when PLL is off): 1: HSE is not crossover fed to the PLL;<br>0: HSI not crossover fed to PLL. | 0 |
| [15:11] | ADCPRE[4:0] | RW | ADC clock source prescaler control:<br>000xx: HB clock is divided by 2 as ADC clock;<br>010xx: HB clock is divided by 4 as ADC clock;<br>100xx: HB clock is divided by 6 as ADC clock;<br>110xx: HB clock is divided by 8 as ADC clock;<br>00100: HB clock is divided by 4 as ADC clock;<br>01100: HB clock is divided by 8 as ADC clock;<br>10100: HB clock is divided by 12 as ADC clock;<br>11100: HB clock is divided by 16 as ADC clock;<br>00101: HB clock is divided by 8 as ADC clock;<br>01101: HB clock is divided by 16 as ADC clock;<br>10101: HB clock is divided by 24 as ADC clock;<br>11101: HB clock is divided by 32 as ADC clock;<br>00110: HB clock is divided by 16 as ADC clock;<br>01110: HB clock is divided by 32 as ADC clock;<br>10110: HB clock is divided by 48 as ADC clock;<br>11110: HB clock is divided by 64 as ADC clock;<br>00111: HB clock is divided by 32 as ADC clock;<br>01111: HB clock is divided by 64 as ADC clock;<br>10111: HB clock is divided by 96 as ADC clock;<br>11111: HB clock is divided by 128 as ADC clock; | 0 |
| [10:4] | HPRE[6:0] | RW | HB clock source pre-frequency division control:<br>0000000: SYSCLK does not divide frequency;<br>0000001: SYSCLK divided by 2;<br>0000010: SYSCLK divided by 3;<br>……<br>1111110: SYSCLK divided by 127;<br>1111111: SYSCLK divided by 128. | 0 |
| [3:2] | SWS[1:0] | RO | System clock (SYSCLK) status (hardware set):<br>00: The system clock source is HSI;<br>01: The system clock source is HSE;<br>10: The system clock source is PLL;<br>11: Not available. | 0 |
| [1:0] | SW[1:0] | RW | Select the system clock source:<br>00: HSI as the system clock;<br>01: HSE as the system clock;<br>10: PLL as the system clock;<br>11: Not available. | 0 |

### 3.4.3 Clock Interrupt Register (RCC_INTR)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | PLL | HSE | HSI | Reser | LSI |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | RDYC | RDYC | RDYC | ved | RDYC |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | PLLRDYIE | HSERDYIE | HSIRDYIE | Reserved | LSIRDYIE | Reserved | | | PLLRDYF | HSERDYF | HSIRDYF | Reserved | LSIRDYF |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:21] | Reserved | RO | Reserved | 0 |
| 20 | PLLRDYC | WO | Clear the PLL ready interrupt flag:<br>1: Clear PLLRDYF interrupt flag;<br>0: No action. | 0 |
| 19 | HSERDYC | WO | Clear the HSE oscillator ready interrupt flag:<br>1. Clear the HSERDYF interrupt flag;<br>0: No action. | 0 |
| 18 | HSIRDYC | WO | Clear the HSI oscillator ready interrupt flag:<br>1. Clear the HSIRDYF interrupt flag;<br>0: No action. | 0 |
| 17 | Reserved | RO | Reserved | 0 |
| 16 | LSIRDYC | WO | Clear the LSI oscillator ready interrupt flag:<br>1. Clear the LSIRDYF interrupt flag;<br>0: No action. | 0 |
| [15:13] | Reserved | RO | Reserved | 0 |
| 12 | PLLRDYIE | RW | PLL ready interrupt enable bits:<br>1: Enable PLL ready interrupt;<br>0: Disable PLL ready interrupt. | 0 |
| 11 | HSERDYIE | RW | HSE ready interrupt enable bits:<br>1: Enable HSE ready interrupt;<br>0: Disable HSE ready interrupt. | 0 |
| 10 | HSIRDYIE | RW | HSI ready interrupt enable bits:<br>1: Enable HSI ready interrupt;<br>0: Disable HSI ready interrupt. | 0 |
| 9 | Reserved | RO | Reserved | 0 |
| 8 | LSIRDYIE | RW | LSI ready interrupt enable bits:<br>1: Enable LSI ready interrupt;<br>0: Disable LSI ready interrupt. | 0 |
| [7:5] | Reserved | RO | Reserved | 0 |
| 4 | PLLRDYF | RO | PLL clock ready lock interrupt flag:<br>1: PLL clock lock generates interrupt;<br>0: No PLL clock lock interrupt.<br>Hardware set, software write PLLRDYC bit 1 cleared. | 0 |
| 3 | HSERDYF | RO | HSE clock ready interrupt sign:<br>1: HSE clock ready to generate interrupt;<br>0: No HSE clock ready interrupt.<br>Hardware set, software write HSERDYC bit 1 cleared. | 0 |
| 2 | HSIRDYF | RO | HSI clock ready interrupt flag:<br>1: HSI clock ready to generate an interrupt;<br>0: No HSI clock ready interrupt.<br>When the hardware is set, the software writes HSIRDYC bit 1 to clear it. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | LSIRDYF | RO | LSI clock ready interrupt flag:<br>1: LSI clock ready to generate interrupt;<br>0: No LSI clock ready interrupt.<br>Hardware set, software write LSIRDYC bit 1 cleared. | 0 |

### 3.4.4 PB2 Peripheral Reset Register (RCC_PB2PRSTR)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | UART1 RST | Reserved | SPI1 RST | TIM1 RST | Reserved | ADC1 RST | Reserved | | | | IOPC RST | IOPB RST | IOPA RST | Reserved | AFIO RST |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved. | 0 |
| 14 | UART1RST | RW | UART1 interface reset control:<br>1: Reset module; 0: No effect. | 0 |
| 13 | Reserved | RO | Reserved. | 0 |
| 12 | SPI1RST | RW | SPI1 interface reset control:<br>1: Reset module; 0: No effect. | 0 |
| 11 | TIM1RST | RW | TIM1 module reset control.<br>1: Reset module; 0: No effect. | 0 |
| 10 | Reserved | RO | Reserved | 0 |
| 9 | ADC1RST | RW | ADC1 module reset control:<br>1: Reset module; 0: No effect. | 0 |
| [8:5] | Reserved | RO | Reserved. | 0 |
| 4 | IOPCRST | RW | PC port module reset control for I/O:<br>1: Reset module; 0: No effect. | 0 |
| 3 | IOPBRST | RW | PB port module reset control for I/O:<br>1: Reset module; 0: No effect. | 0 |
| 2 | IOPARST | RW | PA port module reset control for I/O:<br>1: Reset module; 0: No effect. | 0 |
| 1 | Reserved | RO | Reserved. | 0 |
| 0 | AFIORST | RW | I/O auxiliary function module reset control:<br>1: Reset module; 0: No effect. | 0 |

### 3.4.5 PB1 Peripheral Reset Register (RCC_PB1PRSTR)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | PWR RST | Reserved | | | | | | I2C1 RST | Reserved | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | WWDG RST | Reserved | | | | | | | | | TIM3 RST | TIM2 RST |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:29] | Reserved | RO | Reserved. | 0 |
| 28 | PWRRST | RW | Power interface module reset control:<br>1: Reset module; 0: No effect. | 0 |
| [27:22] | Reserved | RO | Reserved. | 0 |
| 21 | I2C1RST | RW | I2C1 interface reset control: | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Reset module; 0: No effect. | |
| [20:12] | Reserved | RW | Reserved. | 0 |
| 11 | WWDGRST | RW | Window watchdog reset control:<br>1: Reset module; 0: No effect. | 0 |
| [10:2] | Reserved | RO | Reserved. | 0 |
| 1 | TIM3RST | RW | Timer 3 module reset control:<br>1: Reset module; 0: No effect. | 0 |
| 0 | TIM2RST | RW | Timer 2 module reset control:<br>1: Reset module; 0: No effect. | 0 |

### 3.4.6 HB Peripheral Clock Enable Register (RCC_AHBPCENR)

Offset address: 0x14

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | OPAEN | Reserved | USBPD1EN | USBPD0EN | USBFSEN | SRAMEN | Reserved | DMAEN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:8] | Reserved | RO | Reserved | 0 |
| 7 | OPAEN | RW | OPA_CMP module clock enable bit:<br>1: Module clock on;<br>0: Module clock off. | 0 |
| 6 | Reserved | RO | Reserved | 0 |
| 5 | USBPD1EN | RW | USBPD1 clock enable:<br>1: Module clock on;<br>0: Module clock off. | 0 |
| 4 | USBPD0EN | RW | USBPD0 clock enable:<br>1: Module clock on;<br>0: Module clock off. | 0 |
| 3 | USBFSEN | RW | USBFS module clock enable bit:<br>1: Module clock on;<br>0: Module clock off. | 0 |
| 2 | SRAMEN | RW | SRAM interface module clock enable bit:<br>1: SRAM interface module clock on in Sleep mode;<br>0: SRAM interface module clock off in Sleep mode. | 1 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | DMAEN | RW | DMA module clock enable bit:<br>1: Module clock on;<br>0: Module clock off. | 0 |

### 3.4.7 PB2 Peripheral Clock Enable Register (RCC_PB2PCENR)

Offset address: 0x18

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | UART1EN | Reserved | SPI1EN | TIM1EN | Reserved | ADC1EN | Reserved | | | | IOPCEN | IOPBEN | IOPAEN | Reserved | AFIOEN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved. | 0 |
| 14 | UART1EN | RW | UART1 interface clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| 13 | Reserved | RO | Reserved. | 0 |
| 12 | SPI1EN | RW | SPI1 interface clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| 11 | TIM1EN | RW | TIM1 module clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| 10 | Reserved | RO | Reserved. | 0 |
| 9 | ADC1EN | RW | ADC1 module clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| [8:5] | Reserved | RO | Reserved. | 0 |
| 4 | IOPCEN | RW | PC port module clock enable bit for I/O:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| 3 | IOPBEN | RW | PB port module clock enable bit for I/O:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| 2 | IOPAEN | RW | PA port module clock enable bit for I/O:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| 1 | Reserved | RO | Reserved. | 0 |
| 0 | AFIOEN | RW | I/O auxiliary function module clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |

### 3.4.8 PB1 Peripheral Clock Enable Register (RCC_PB1PCENR)

Offset address: 0x1C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | PWREN | Reserved | | | | | | I2C1EN | Reserved | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | WWDGEN | Reserved | | | | | | | | | TIM3EN | TIM2EN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:29] | Reserved | RO | Reserved. | 0 |
| 28 | PWREN | RW | Power interface module clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| [27:22] | Reserved | RO | Reserved. | 0 |
| 21 | I2C1EN | RW | I2C1 interface clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| [20:12] | Reserved | RO | Reserved. | 0 |
| 11 | WWDGEN | RW | Window watchdog clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| [10:2] | Reserved | RO | Reserved. | 0 |
| 1 | TIM3EN | RW | Timer 3 module clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |
| 0 | TIM2EN | RW | Timer 2 module clock enable bit:<br>1: Module clock is on; 0: Module clock is off. | 0 |

### 3.4.9 Control/Status Register (RCC_RSTSCKR)

Offset address: 0x24

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPWRRSTF | WWDGRSTF | Reserved | SFTRSTF | PORRSTF | PINRSTF | ADC_AWDRSTF | RMVF | USBPD1RSTF | USBPD0RSTF | Reserved | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | LSIRDY | LSION |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31 | LPWRRSTF | RW | Low-power reset flag:<br>1: Occurrence of low-power resets.<br>0: No low-power reset occurs.<br>Set to 1 by hardware when a low-power management reset occurs; cleared by software writing of the RMVF bit. | 0 |
| 30 | WWDGRSTF | RW | Window watchdog reset flag:<br>1: Occurrence of a window watchdog reset.<br>0: No window watchdog reset occurs.<br>Set to 1 by hardware when a window watchdog reset occurs; cleared by software writing of the RMVF bit. | 0 |
| 29 | Reserved | RO | Reserved | 0 |
| 28 | SFTRSTF | RW | Software reset flag:<br>1: Software reset occurs.<br>0: No software reset occurs.<br>Set to 1 by hardware when a software reset occurs; software write RMVF bit cleared. | 0 |
| 27 | PORRSTF | RW | Power-up/power-down reset flag:<br>1: Power-up/power-down reset occurs.<br>0: No power-up/power-down reset occurs.<br>Set to 1 by hardware when power-up/power-down reset occurs; cleared by software writing of RMVF bit. | 1 |
| 26 | PINRSTF | RW | External manual reset (NRST pin) flag:<br>1: Occurrence of NRST pin reset.<br>0: No NRST pin reset occurs.<br>Set to 1 by hardware when NRST pin reset occurs; cleared by software writing of RMVF bit. | 0 |
| 25 | ADC_AWDRSTF | RO | ADC module analog watchdog reset flag:<br>1: ADC analog watchdog reset occurs;<br>0: No ADC analog watchdog reset occurs.<br>Cleared by software writing of RMVF bit. | 0 |
| 24 | RMVF | RW | Clear reset flag control:<br>1: Clear the reset flag.<br>0: No effect. | 0 |
| 23 | USBPD1RSTF | RW | USBPD1 reset flag:<br>1: USBPD1 reset occurs;<br>0: No USBPD1 reset occurs.<br>Set to 1 by hardware when USBPD1 pin reset occurs; cleared by software writing of RMVF bit. | 0 |
| 22 | USBPD0RSTF | RW | USBPD0 reset flag:<br>1: USBPD0 reset occurs;<br>0: No USBPD0 reset occurs.<br>Set to 1 by hardware when USBPD0 pin reset occurs; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | cleared by software writing of RMVF bit. | |
| [21:2] | Reserved | RO | Reserved. | 0 |
| 1 | LSIRDY | RO | Internal low-speed clock (LSI) stability ready flag (set by hardware):<br>1. Internal low-speed clock (340kHz) is stable;<br>0: Internal low-speed clock (340kHz) is not stable.<br>*Note: After the LSION bit is cleared, it takes 3 LSI cycles to clear it.* | 0 |
| 0 | LSION | RW | Internal low-speed clock (LSI) enable control bit:<br>1: Enable LSI(340kHz) oscillator;<br>0: Disable LSI (340kHz) oscillator. | 0 |

*Note: Except for the reset flag which can only be cleared by power-on reset, others are cleared by system Reset.*

### 3.4.10 HB Peripheral Reset Register (RCC_AHBRSTR)

Offset address: 0x28

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | OPARST | Reserved | USBPD1RST | USBPD0RST | USBFSRST | Reserved | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:8] | Reserved | RO | Reserved. | 0 |
| 7 | OPARST | RW | OPA_CMP module reset control:<br>1: Reset module;<br>0: No effect. | 0 |
| 6 | Reserved | RO | Reserved. | 0 |
| 5 | USBPD1RST | RW | USBPD1 module reset control:<br>1: Reset module;<br>0: No effect. | 0 |
| 4 | USBPD0RST | RW | USBPD0 module reset control:<br>1: Reset module;<br>0: No effect. | 0 |
| 3 | USBFSRST | RW | USBFS module reset control:<br>1: Reset module;<br>0: No effect. | 0 |
| [2:0] | Reserved | RO | Reserved. | 0 |

# Chapter 4 Window Watchdog (WWDG)

Window Watchdog is generally used to monitor system operation for software faults such as external disturbances, unforeseen logic errors, and other conditions. It requires a counter refresh (Dog feeding) within a specific window time (With upper and lower limits), otherwise earlier or later than this window time the watchdog circuit will generate a system Reset.

## 4.1 Main Features

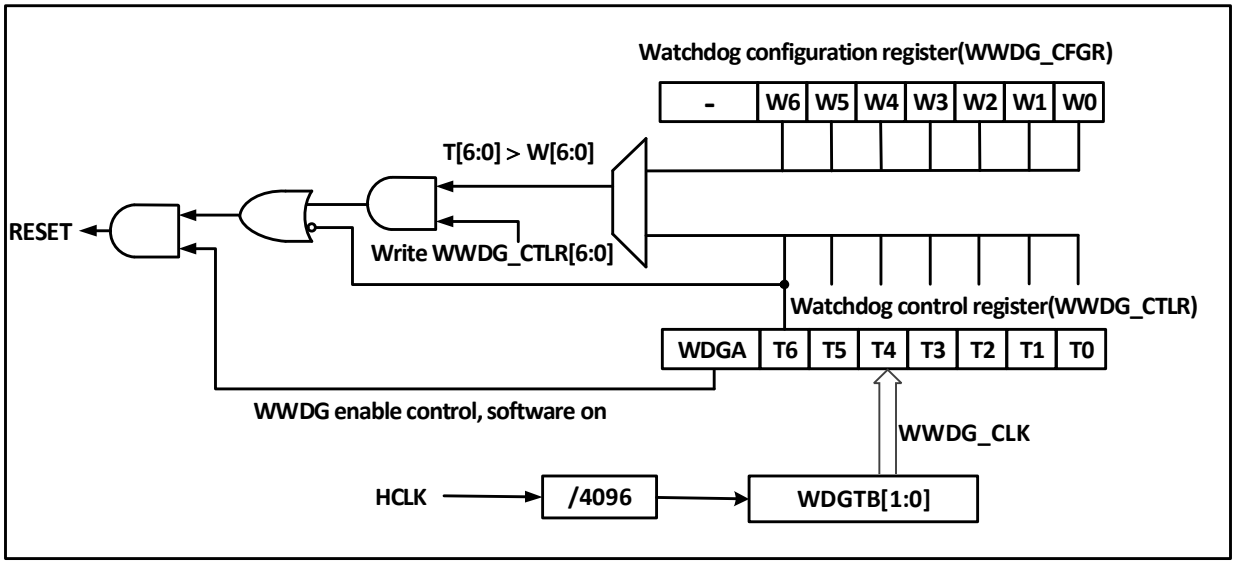
- Programmable 7-bit downcounter
- Biconditional reset: the downcounter value is less than 0x40, or the counter value is reloaded outside the window time
- Early Wake-up Notification (EWI) function for timely dog feeding action to prevent system Reset

## 4.2 Function Description

### 4.2.1 Principle and Application

The window watchdog operation is based on a 7-bit downcounter, which is mounted under the HB bus and counts the dividing frequency of the time base WWDG_CLK source (HCLK/4096) clock with the dividing factor set in the WDGTB[1:0] field in the configuration register WWDG_CFGR. The downcounter is in the free-running state, and the counter keeps cycling downcount regardless of whether the watchdog function is on or not. As shown in Figure 6-1, the block diagram of the internal structure of the window watchdog.

Figure 4-1 Block diagram of Window Watchdog structure




- Enable Window Watchdog

After a system Reset, the watchdog is off. Setting the WDGA bit of the WWDG_CTLR register enables the watchdog, and then it cannot be turned off again unless a reset occurs.

*Note: The watchdog function can be stopped indirectly by setting the RCC_PB1PCENR register to turn off the clock source of WWDG and suspend the WWDG_CLK count, or by setting the RCC_PB1PRSTR register to reset the WWDG module, which is equivalent to the role of reset.*

- Watchdog Configuration

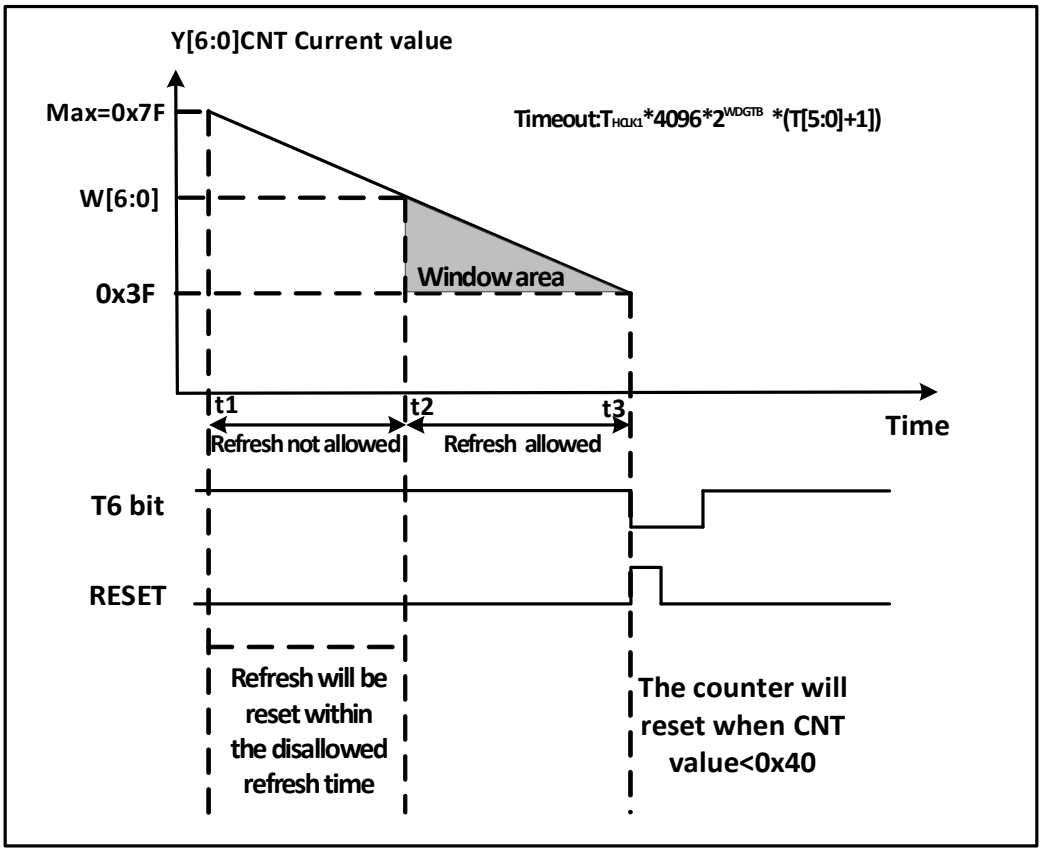
The watchdog is internally a 7-bit counter that runs in a continuous decreasing cycle and supports read and write access. To use the watchdog reset function, the following actions need to be performed.

1) Counting time base: via the WDGTB[1:0] bit field of the WWDG_CFGR register, note that the WWDG module clock of the RCC unit should be turned on.
2) Window counter: Set the W[6:0] bit field of WWDG_CFGR register, this counter is used by hardware as a comparison with the current counter, the value is configured by user software and will not change. It is used as the upper limit value of the window time.
3) Watchdog enable: WDG_CTLR register WDGA bit software set to 1, to turn on the watchdog function, you can system reset.
4) Feed the dog: i.e., refresh the current counter value and configure the T[6:0] bit field of the WWDG_CTLR register. This action needs to be executed within the periodic window time after the watchdog function is turned on, otherwise a watchdog reset action will occur.

- Dog feeding window time

As shown in Figure 6-2, the gray area is the monitoring window area of the window watchdog, whose upper time t2 corresponds to the point in time when the current counter value reaches the window value W[6:0]; its lower time t3 corresponds to the point in time when the current counter value reaches 0x3F. This area time t2<t<t3 can be fed with a dog operation (Write T[6:0]) to refresh the current counter value.

- Figure 4-2 Counting mode of Window Watchdog




- Watchdog reset

1) When the value of T[6:0] counter changes from 0x40 to 0x3F due to no timely dog feeding operation, a "window watchdog reset" will occur and a system reset will be generated. That is, the T6-bit is detected as 0 by the hardware and a system reset will occur.
   *Note: The application can write T6-bit to 0 by software to achieve system Reset, which is equivalent to software reset function.*
2) When the counter refresh action is executed within the disallowed dog feeding time, i.e., the write T[6:0] bit field is operated within $t_1 \leqslant t \leqslant t_2$ time, a 'window watchdog reset' will occur and a system Reset will be generated.

- Wake up in advance

To prevent the system Reset caused by not refreshing the counter in time, the watchdog module provides an early wakeup interrupt (EWI) notification. When the counter self-decreases to 0x40, an early wake-up signal is generated and the EWIF flag is set to 1. If the EWI bit is set, a window watchdog interrupt will be triggered at the same time. At this time, there is 1 counter clock cycle (Self-decrement to 0x3F) before the hardware reset, and the application can perform the dog feeding operation instantly within this time.

### 4.2.2 Debug Mode

When the system enters Debug mode, the counter of WWDG can be configured by the debug module register to continue or stop.

## 4.3 Register Description

Table 4-1 WWDG-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_WWDG_CTLR | 0x40002C00 | Control register | 0x007F |
| R16_WWDG_CFGR | 0x40002C04 | Configuration Register | 0x007F |
| R16_WWDG_STATR | 0x40002C08 | Status Register | 0x0000 |

### 4.3.1 Control Register (WWDG_CTLR)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved | 0 |
| 7 | WDGA | RW1 | Window watchdog reset enable bit.<br>1: Turn on the watchdog function (Which generates a reset signal).<br>0: Disable the watchdog function. Software write 1 is on, but only allows hardware to clear 0 after reset. | 0 |
| [6:0] | T[6:0] | RW | The 7-bit self-decrement counter decrements by 1 every $4096*2^{WDGTB}$ HCLK cycles. A watchdog reset is generated when the counter decrements from 0x40 to 0x3F, i.e., when T6 jumps to 0. | 0x7F |

### 4.3.2 Configuration Register (WWDG_CFGR)

Offset address: 0x04

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:10] | Reserved | RO | Reserved | 0 |
| 9 | EWI | RW1 | Early wakeup interrupt enable bit.<br>If this position is 1, an interrupt is generated when the counter value reaches 0x40. This bit can only be invited to 0 by hardware after a reset. | 0 |
| [8:7] | WDGTB[1:0] | RW | Window watchdog clock division selection.<br>00: Divided by 1, counting time base = HCLK/4096.<br>01: Divided by 2, counting time base = HCLK /4096/2.<br>10: Divided by 4, counting time base = HCLK /4096/4.<br>11: Divided by 8, counting time base = HCLK /4096/8. | 0 |
| [6:0] | W[6:0] | RW | Window watchdog 7-bit window value. Used to compare with the counter value. The feed dog operation can only be performed when the counter value is less than the window value and greater than 0x3F. | 0x7F |

### 4.3.3 Status Register (WWDG_STATR)

Offset address: 0x08

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:1] | Reserved | WO | Reserved | 0 |
| 0 | EWIF | RW0 | Wake up the interrupt flag bit early.<br>When the counter reaches 0x40, this bit is set in | 0 |

| | | | | |
|---|---|---|---|---|
| | | | hardware and must be cleared to 0 by software; the user setting is invalid. Even if the EWI is not set, this bit will still be set as usual when the event occurs. | |

# Chapter 5 Interrupt and Events (PFIC)

The CH32M030 series has a built-in Programmable Fast Interrupt Controller (PFIC) that supports up to 255 interrupt vectors. The current system manages 31 peripheral interrupt channels and 5 core interrupt channels, the others are reserved.

## 5.1 Main Features

- 31 peripheral interrupts, each interrupt request has independent trigger and mask bits, status bits
- 2 levels of nested interrupt entry and exit hardware automatic stack and resume, no instruction overhead
- 4 programmable fast interrupt channels with customizable interrupt vector addresses
- 5 non-maskable interrupts

## 5.2 System Timer

- CH32M030 Series

RISC-V3B core comes with a 32-bit addition and subtraction counter (SysTick), which supports HCLK or HCLK/8 as the time base, which has high priority and can be used as the time reference after calibration.

## 5.3 Vector Table of Interrupts and Exceptions

Table 5-1 CH32M030 series vector table

| No. | Priority | Type | Name | Description | Entrance address |
|---|---|---|---|---|---|
| 0 | - | - | - | - | 0x00000000 |
| 1 | - | - | - | - | 0x00000004 |
| 2 | -5 | Fixed | NMI | Non-maskable interrupts | 0x00000008 |
| 3 | -4 | Fixed | EXC | Abnormal interruptions | 0x0000000C |
| 4 | - | - | - | - | 0x00000010 |
| 5 | -3 | Fixed | ECALL-M | Machine mode callback interrupt | 0x00000014 |
| 6-7 | - | - | - | - | 0x00000018-0x0000001C |
| 8 | -2 | Fixed | ECALL-U | User mode callback interrupt | 0x00000020 |
| 9 | -1 | Fixed | BREAKPOINT | Breakpoint callback interruption | 0x00000024 |
| 10-11 | - | - | - | - | 0x00000028-0x0000002C |
| 12 | 0 | Programmable | SysTick | System timer interrupt | 0x00000030 |
| 13 | - | - | - | - | 0x00000034 |
| 14 | 1 | Programmable | SWI | Software interrupt | 0x00000038 |
| 15 | - | - | - | - | 0x0000003C |
| 16 | 2 | Programmable | WWDG | Window timer interrupt | 0x00000040 |
| 17 | 3 | Programmable | OPA | OPA global interrupt | 0x00000044 |
| 18 | 4 | Programmable | FLASH | FLASH global interrupt | 0x00000048 |
| 19 | 5 | Programmable | RCC | Reset and clock control interrupt | 0x0000004C |
| 20 | 6 | Programmable | EXTI7_0 | EXTI line 0-7 interrupt | 0x00000050 |
| 21 | 7 | Programmable | AWU | AWU interrupt | 0x00000054 |
| 22 | 8 | Programmable | DMA1_CH1 | DMA1 channel 1 global interrupt | 0x00000058 |
| 23 | 9 | Programmable | DMA1_CH2 | DMA1 channel 2 global interrupt | 0x0000005C |
| 24 | 10 | Programmable | DMA1_CH3 | DMA1 channel 3 global interrupt | 0x00000060 |

| | | | | | |
|---|---|---|---|---|---|
| 25 | 11 | Programmable | DMA1_CH4 | DMA1 channel 4 global interrupt | 0x00000064 |
| 26 | 12 | Programmable | DMA1_CH5 | DMA1 channel 5 global interrupt | 0x00000068 |
| 27 | 13 | Programmable | DMA1_CH6 | DMA1 channel 6 global interrupt | 0x0000006C |
| 28 | 14 | Programmable | DMA1_CH7 | DMA1 channel 7 global interrupt | 0x00000070 |
| 29 | 15 | Programmable | ADC | ADC global interrupt | 0x00000074 |
| 30 | 16 | Programmable | I2C1_EV | I2C1 event interrupt | 0x00000078 |
| 31 | 17 | Programmable | I2C1_ER | I2C1 error interrupt | 0x0000007C |
| 32 | 18 | Programmable | UART1 | UART1 global interrupt | 0x00000080 |
| 33 | 19 | Programmable | EXTI15_8 | EXTI line 8-15 interrupt | 0x00000084 |
| 34 | 20 | Programmable | TIM1BRK | TIM1 brake interrupt | 0x00000088 |
| 35 | 21 | Programmable | TIM1UP | TIM1 update interrupt | 0x0000008C |
| 36 | 22 | Programmable | TIM1TRG | TIM1 trigger interrupt | 0x00000090 |
| 37 | 23 | Programmable | TIM1CC | TIM1 capture compare interrupt | 0x00000094 |
| 38 | 24 | Programmable | TIM2 | TIM2 global interrupt | 0x00000098 |
| 39 | 25 | Programmable | USBPD0 | USBPD0 global interrupt | 0x0000009C |
| 40 | 26 | Programmable | USBPD0WakeUP | USBPD0 Wake-up interrupt | 0x000000A0 |
| 41 | 27 | Programmable | USBPD1 | USBPD1 global interrupt | 0x000000A4 |
| 42 | 28 | Programmable | USBPD1WakeUP | USBPD1 Wake-up interrupt | 0x000000A8 |
| 43 | 29 | Programmable | TIM3 | TIM3 global interrupt | 0x000000AC |
| 44 | 30 | Programmable | SPI1 | SPI1 global interrupt | 0x000000B0 |
| 45 | 31 | Programmable | USBFS | USBFS global interrupt | 0x000000B4 |
| 46 | 32 | Programmable | USBFSWakeUP | USBFS Wake-up interrupt | 0x000000B8 |

*Note: 1. NMI, EXC, ECALL-M, ECALL-U, BREAKPOINT are enabled by default;*

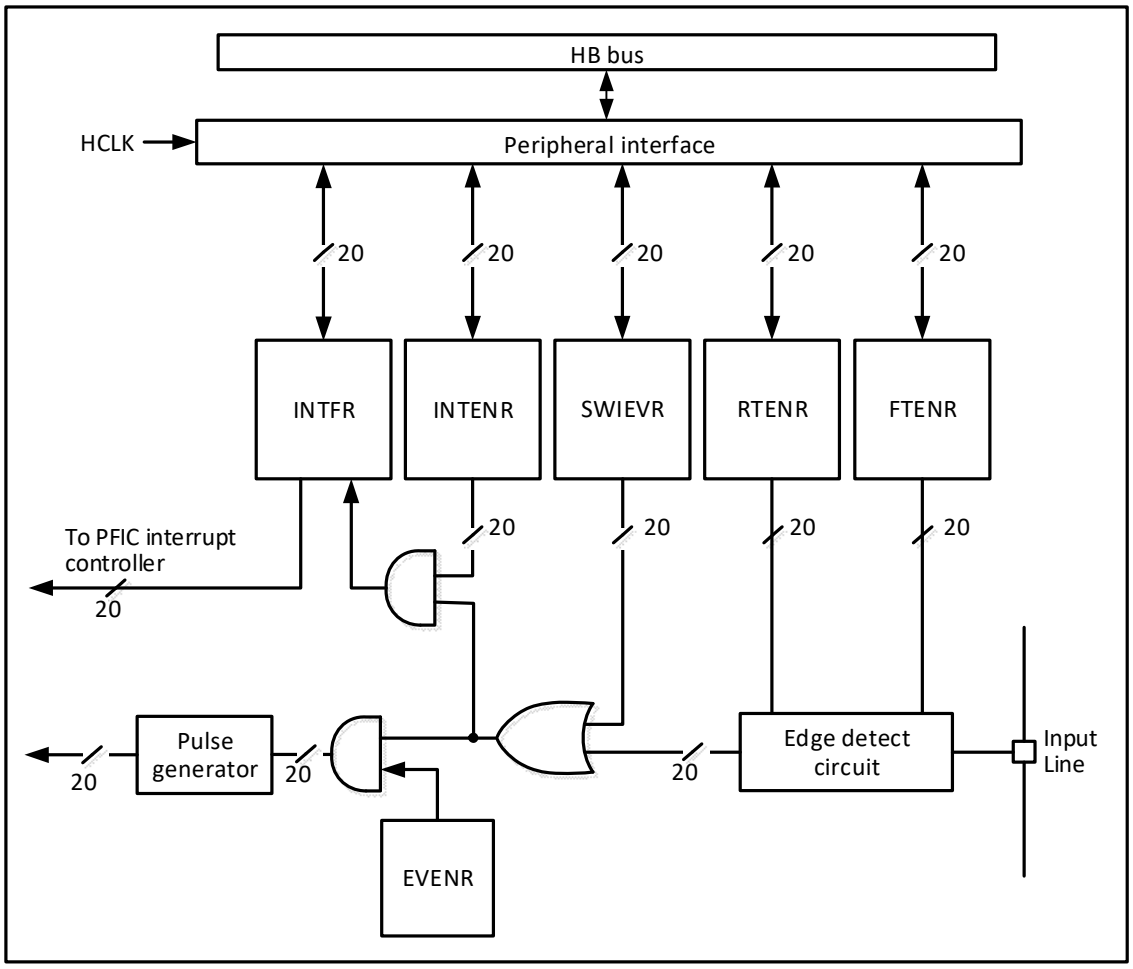
*2. ECALL-M、ECALL-U、BREAKPOINT is a case of EXC;*

*3. NMI, EXC, ECALL-M, ECALL-U, BREAKPOINT support interrupt hang clear and set operation, do not support interrupt enable clear and set operation.*

# 5.4 External Interrupt and Event Controller (EXTI)

## 5.4.1 Overview

Figure 5-1 External interrupt (EXTI) interface block diagram




As can be seen from Figure 5-1, the trigger source of the external interrupt can be either a software interrupt (SWIEVR) or an actual external interrupt channel. The signal of the external interrupt channel will be screened by the edge detect circuit first. Whenever one of the software interrupt or external interrupt signals is generated, it will be output to 2 AND gate circuits, event enable and interrupt enable, through the OR gate circuit in the figure, as long as an interrupt is enabled or an event is enabled, an interrupt or an event will be generated. 6 registers of EXTI are accessed by the processor through the HB interface.

## 5.4.2 Wake-up Event

The system can wake up the Sleep mode caused by the WFE command through a wake-up event. The wake-up event is generated by either of the following two configurations.

- Enable an interrupt in a peripheral register, but not enabling this interrupt in the PFIC of the core, and enabling the SEVONPEND bit in the core at the same time. Embodied in EXTI, it is to enable an EXTI interrupt, but not to enable the EXTI interrupt in PFIC, and to enable the SEVONPEND bit at the same time. When the CPU wakes up from WFE, it needs to clear the EXTI interrupt flag bit and the PFIC pending bit.
- Enable an EXTI channel as an event channel eliminates the need for the CPU to clear the interrupt flag bit and the PFIC pending bit after waking up from the WFE.

### 5.4.3 Description

Using an external interrupt requires configuring the corresponding external interrupt channel, i.e. selecting the corresponding trigger edge and enabling the corresponding interrupt. When the set trigger edge appears on the external interrupt channel, an interrupt request will be generated and the corresponding interrupt flag bit will be set. The flag bit can be cleared by writing 1 to the flag bit.

Steps for using external hardware interrupts.

1) Configuration of GPIO operations.
2) Configure the interrupt enable bit (EXTI_INTENR) for the corresponding external interrupt channel.
3) Configuring the trigger edge (EXTI_RTENR or EXTI_FTENR) to select rising edge trigger, falling edge trigger or double edge trigger.
4) Configure EXTI interrupts in the core's PFIC to ensure they can respond correctly.

Steps for using external hardware events.

1) Configuration of GPIO operations.
2) Configure the event enable bit (EXTI_EVENR) for the corresponding external interrupt channel.
3) Configure the trigger edge (EXTI_RTENR or EXTI_FTENR) to select rising edge trigger, falling edge trigger, or double edge trigger.

Using the software interrupt/event steps.

1) Enabling external interrupts (EXTI_INTENR) or external events (EXTI_EVENR).
2) If using interrupt service functions, the EXTI interrupt needs to be set in the core's PFIC.
3) Set the software interrupt trigger (EXTI_SWIEVR), that is, an interrupt will be generated.

### 5.4.4 External Event Mapping

Table 5-2 EXTI Interrupt Mapping

| External interrupt/ Event lines | Mapping event description |
|---|---|
| EXTI0~EXTI15 | Px0~Px15 (x=A/B/C), any IO port can enable external interrupt/event function, which is configured by AFIO_EXTICRx register. |
| EXTI16 | USBPD0 wake-up event |
| EXTI17 | AWU event |
| EXTI18 | USBPD1 wake-up event |
| EXTI19 | USBFS wake-up event |

## 5.5 Register Description

### 5.5.1 EXTI Registers

Table 5-3 EXTI-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_EXTI_INTENR | 0x40010400 | Interrupt enable register | 0x00000000 |
| R32_EXTI_EVENR | 0x40010404 | Event enable register | 0x00000000 |
| R32_EXTI_RTENR | 0x40010408 | Rising edge trigger enable register | 0x00000000 |
| R32_EXTI_FTENR | 0x4001040C | Falling edge trigger enable register | 0x00000000 |
| R32_EXTI_SWIEVR | 0x40010410 | Soft interrupt event register | 0x00000000 |
| R32_EXTI_INTFR | 0x40010414 | Interrupt flag register | 0x0000XXXX |

#### 5.5.1.1 Interrupt Enable Register (EXTI_INTENR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | MR19 | MR18 | MR17 | MR16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MR15 | MR14 | MR13 | MR12 | MR11 | MR10 | MR9 | MR8 | MR7 | MR6 | MR5 | MR4 | MR3 | MR2 | MR1 | MR0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:20] | Reserved | RO | Reserved | 0 |
| [19:0] | MRx | RW | Enable the interrupt request signal for external interrupt channel x.<br>1: Enable interrupts for this channel.<br>0: Mask interrupts for this channel. | 0 |

#### 5.5.1.2 Event Enable Register (EXTI_EVENR)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | MR19 | MR18 | MR17 | MR16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MR15 | MR14 | MR13 | MR12 | MR11 | MR10 | MR9 | MR8 | MR7 | MR6 | MR5 | MR4 | MR3 | MR2 | MR1 | MR0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:20] | Reserved | RO | Reserved. | 0 |
| [19:0] | MRx | RW | Enable the event request signal for external interrupt channel x.<br>1: Enable events for this channel.<br>0: Mask events for this channel. | 0 |

#### 5.5.1.3 Rising Edge Trigger Enable Register (EXTI_RTENR)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | TR19 | TR18 | TR17 | TR16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TR15 | TR14 | TR13 | TR12 | TR11 | TR10 | TR9 | TR8 | TR7 | TR6 | TR5 | TR4 | TR3 | TR2 | TR1 | TR0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:20] | Reserved | RO | Reserved. | 0 |
| [19:0] | TRx | RW | Enable rising edge triggering for external interrupt channel x.<br>1: Enable rising edge triggering for this channel. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0: Disable rising edge triggering for this channel. | |

### 5.5.1.4 Falling Edge Trigger Enable Register (EXTI_FTENR)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | TR19 | TR18 | TR17 | TR16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TR15 | TR14 | TR13 | TR12 | TR11 | TR10 | TR9 | TR8 | TR7 | TR6 | TR5 | TR4 | TR3 | TR2 | TR1 | TR0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:20] | Reserved | RO | Reserved | 0 |
| [19:0] | TRx | RW | Enable falling edge triggering for external interrupt channel x.<br>1: Enable falling edge triggering for this channel.<br>0: Disable falling edge triggering for this channel. | 0 |

### 5.5.1.5 Software Interrupt Event Register (EXTI_SWIEVR)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | SWIER 19 | SWIER 18 | SWIER 17 | SWIER 16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWIER15 | SWIER 14 | SWIER13 | SWIER12 | SWIER11 | SWIER 10 | SWIER 9 | SWIER 8 | SWIER 7 | SWIER 6 | SWIER 5 | SWIER 4 | SWIER 3 | SWIER 2 | SWIER 1 | SWIER 0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:20] | Reserved | RO | Reserved | 0 |
| [19:0] | SWIERx | RW | A software interrupt is set on the corresponding externally triggered interrupt channel. Setting it here causes the interrupt flag bit (EXTI_INTFR) to correspond to the position bit, and if interrupt enable (EXTI_INTENR) or event enable (EXTI_EVENR) is on, then an interrupt or event will be generated. | 0 |

### 5.5.1.6 Interrupt Flag Register (EXTI_INTFR)

Offset address: 0x14

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | IF19 | IF18 | IF17 | IF16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IF15 | IF14 | IF13 | IF12 | IF11 | IF10 | IF9 | IF8 | IF7 | IF6 | IF5 | IF4 | IF3 | IF2 | IF1 | IF0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:20] | Reserved | RO | Reserved | 0 |
| [19:0] | IFx | W1 | The interrupt flag bit, this location bit flags that a | X |

| | | | | |
|---|---|---|---|---|
| | | | corresponding external interrupt has occurred. A write of 1 clears this bit. | |

### 5.5.2 PFIC Registers

Table 5-4 PFIC-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_PFIC_ISR1 | 0xE000E000 | PFIC interrupt enable status register 1 | 0x0000032C |
| R32_PFIC_ISR2 | 0xE000E004 | PFIC interrupt enable status register 2 | 0x00000000 |
| R32_PFIC_IPR1 | 0xE000E020 | PFIC interrupt pending status register 1 | 0x00000000 |
| R32_PFIC_IPR2 | 0xE000E024 | PFIC interrupt pending status register 2 | 0x00000000 |
| R32_PFIC_ITHRESDR | 0xE000E040 | PFIC interrupt priority threshold configuration register | 0x00000000 |
| R32_PFIC_GISR | 0xE000E04C | PFIC interrupt global status register | 0x00000000 |
| R32_PFIC_VTFIDR | 0xE000E050 | PFIC VTF interrupt ID configuration register | 0xXXXXXXXX |
| R32_PFIC_VTFADDRR0 | 0xE000E060 | PFIC VTF interrupt 0 offset address register | 0xXXXXXXX0 |
| R32_PFIC_VTFADDRR1 | 0xE000E064 | PFIC VTF interrupt 1 offset address register | 0xXXXXXXX0 |
| R32_PFIC_VTFADDRR2 | 0xE000E068 | PFIC VTF interrupt 2 offset address register | 0xXXXXXXX0 |
| R32_PFIC_VTFADDRR3 | 0xE000E06C | PFIC VTF interrupt 3 offset address register | 0xXXXXXXX0 |
| R32_PFIC_IENR1 | 0xE000E100 | PFIC interrupt enable setting register 1 | 0x00000000 |
| R32_PFIC_IENR2 | 0xE000E104 | PFIC interrupt enable setting register 2 | 0x00000000 |
| R32_PFIC_IRER1 | 0xE000E180 | PFIC interrupt enable clear register 1 | 0x00000000 |
| R32_PFIC_IRER2 | 0xE000E184 | PFIC interrupt enable clear register 2 | 0x00000000 |
| R32_PFIC_IPSR1 | 0xE000E200 | PFIC interrupt pending setting register 1 | 0x00000000 |
| R32_PFIC_IPSR2 | 0xE000E204 | PFIC interrupt pending setting register 2 | 0x00000000 |
| R32_PFIC_IPRR1 | 0xE000E280 | PFIC interrupt pending clear register 1 | 0x00000000 |
| R32_PFIC_IPRR2 | 0xE000E284 | PFIC interrupt pending clear register 2 | 0x00000000 |
| R32_PFIC_IACTR1 | 0xE000E300 | PFIC interrupt activation status register 1 | 0x00000000 |
| R32_PFIC_IACTR2 | 0xE000E304 | PFIC interrupt activation status register 2 | 0x00000000 |
| R32_PFIC_IPRIORx | 0xE000E400 | PFIC interrupt priority configuration register | 0x00000000 |
| R32_PFIC_SCTLR | 0xE000ED10 | PFIC system control register | 0x00000000 |

*Note: When masking arbitrary interrupts using the PFIC_IENRx register or global interrupts using the CSR register, add a "fence.i" instruction for synchronization between the kernel control state and the interrupt enable state.*

#### 5.5.2.1 PFIC Interrupt Enable Status Register 1 (PFIC_ISR1)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTENSTA[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTENSTA15 | INTENSTA14 | INTENSTA13 | INTENSTA12 | Reserved | | INTENSTA9 | INTENSTA8 | Reserved | | INTENSTA5 | Reserved | INTENSTA3 | INTENSTA2 | Reserved | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | INTENSTA | RO | 12#-31# interrupt current enable status.<br>1: Current numbered interrupt is enabled.<br>0: Current numbered interrupt is not enabled. | 0 |
| [11:10] | Reserved | RO | Reserved. | 0 |
| [9:8] | INTENSTA | RO | 8#-9# interrupt current enable status. | 11b |

| | | | | |
|---|---|---|---|---|
| | | | 1: Current numbered interrupt is enabled.<br>0: Current numbered interrupt is not enabled. | |
| [7:6] | Reserved | RO | Reserved. | 0 |
| 5 | INTENSTA | RO | 5# interrupt current enable status.<br>1: Current numbered interrupt is enabled.<br>0: Current numbered interrupt is not enabled. | 1 |
| 4 | Reserved | RO | Reserved. | 0 |
| [3:2] | INTENSTA | RO | 2#-3# interrupt current enable status.<br>1: Current numbered interrupt is enabled.<br>0: Current numbered interrupt is not enabled. | 11b |
| [1:0] | Reserved | RO | Reserved. | 0 |

### 5.5.2.2 PFIC Interrupt Enable Status Register 2 (PFIC_ISR2)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | INTENSTA[46:32] | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved. | 0 |
| [14:0] | INTENSTA | RO | 32#-46# interrupt current enable status.<br>1: Current numbered interrupt is enabled.<br>0: Current numbered interrupt is not enabled. | |

### 5.5.2.3 PFIC Interrupt Pending Status Register 1 (PFIC_IPR1)

Offset address: 0x20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PENDSTA[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PENDSTA15 | PENDSTA14 | PENDSTA13 | PENDSTA12 | Reserved | | PENDSTA9 | PENDSTA8 | Reserved | | PENDSTA5 | Reserved | PENDSTA3 | PENDSTA2 | Reserved | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | PENDSTA | RO | 12#-31# interrupts current pending status:<br>1: Current numbered interrupt is pending;<br>0: Current numbered interrupt is not pending. | 0 |
| [11:10] | Reserved | RO | Reserved. | 0 |
| [9:8] | PENDSTA | RO | 8#-9# interrupts current pending status:<br>1: Current numbered interrupt is pending;<br>0: Current numbered interrupt is not pending. | 0 |
| [7:6] | Reserved | RO | Reserved. | 0 |
| 5 | PENDSTA | RO | 5# interrupts current pending status:<br>1: Current numbered interrupt is pending;<br>0: Current numbered interrupt is not pending. | 0 |
| 4 | Reserved | RO | Reserved. | 0 |
| [3:2] | PENDSTA | RO | 2#-3# interrupts current pending status: | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Current numbered interrupt is pending;<br>0: Current numbered interrupt is not pending. | |
| [1:0] | Reserved | RO | Reserved. | 0 |

### 5.5.2.4 PFIC Interrupt Pending Status Register 2 (PFIC_IPR2)

Offset address: 0x24

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | PENDSTA[46:32] | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved. | 0 |
| [14:0] | PENDSTA | RO | 32#-46# interrupts current pending status:<br>1: Current numbered interrupt is pending;<br>0: Current numbered interrupt is not pending. | 0 |

### 5.5.2.5 PFIC Interrupt Priority Threshold Configuration Register (PFIC_ITHRESDR)

Offset address: 0x40

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | THRESHOLD[7:0] | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:8] | Reserved | RO | Reserved. | 0 |
| [7:0] | THRESHOLD[7:0] | RW | Interrupt priority threshold setting value.<br>The interrupt priority value lower than the current setting value, when hung, does not perform interrupt service; this register is 0 means the threshold register function is invalid.<br>[7:6]: Priority threshold;<br>[5:0]: Reserved, fixed to 0, write invalid. | 0 |

### 5.5.2.6 PFIC Interrupt Global Status Register (PFIC_GISR)

Offset address: 0x4C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | LOCKSTA | DBGMODE | GLOBLIE | Reserved | GPENDSTA | GACTSTA | NESTSTA[7:0] | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:14] | Reserved | RO | Reserved | 0 |
| 13 | LOCKSTA | RO | Whether the current processor is in the | 0 |

| | | | | |
|---|---|---|---|---|
| | | | locked state:<br>1. Locked state;<br>0: Not locked state. | |
| 12 | DBGMODE | RO | Whether the current processor is in debugging state:<br>1: Debug state;<br>0: Not debug state. | 0 |
| 11 | GLOBLIE | RO | Global interrupt enable:<br>1: Enable interrupt;<br>0: Disable Interrupt. | 0 |
| 10 | Reserved | RO | Reserved | 0 |
| 9 | GPENDSTA | RO | Whether the current interrupt is in pending state:<br>1: Yes; 0: No. | 0 |
| 8 | GACTSTA | RO | Whether there is an interrupt currently being executed:：<br>1: Yes; 0: No. | 0 |
| [7:0] | NESTSTA[7:0] | RO | The current nesting status is interrupted, and 2-level nesting is currently supported. [3:0] is valid.<br>0x0F: Interrupted in Level 4; (overflow)<br>0x07: Interrupted in Level 3; (overflow)<br>0x03: Interrupted in Level 2;<br>0x01: Interrupted in Level 1;<br>0x00No interruption occurred;<br>Others: Reserved. | 0 |

### 5.5.2.7 PFIC VTF Interrupt ID Configuration Register (PFIC_VTFIDR)

Offset address: 0x50

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VTFID3[7:0] | | | | | | | | VTFID2[7:0] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VTFID1[7:0] | | | | | | | | VTFID0[7:0] | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | VTFID3[7:0] | RW | Configure the interrupt number of VTF interrupt 3. | X |
| [23:16] | VTFID2[7:0] | RW | Configure the interrupt number of VTF interrupt 2. | X |
| [15:8] | VTFID1[7:0] | RW | Configure the interrupt number of VTF interrupt 1. | X |
| [7:0] | VTFID0[7:0] | RW | Configure the interrupt number of VTF interrupt 0. | X |

### 5.5.2.8 PFIC VTF Interrupt 0 Address Register (PFIC_VTFADDRR0)

Offset address: 0x60

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR0[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR0[15:1] | | | | | | | | | | | | | | | VTF0E |

| | |
|---|---|
| | N |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:1] | ADDR0 | RW | VTF interrupt 0 service program address bit[31:1], bit0 is 0. | 0 |
| 0 | VTF0EN | RW | VTF interrupt 0 enable bit.<br>1: Enable VTF interrupt 0 channel;<br>0: off. | 0 |

### 5.5.2.9 PFIC VTF Interrupt 1 Address Register (PFIC_VTFADDRR1)

Offset address: 0x64

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR1[31:16] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR1[15:1] | | | | | | | | | | | | | | | VTF1EN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:1] | ADDR1 | RW | VTF interrupt 1 service program address bit[31:1]. | X |
| 0 | VTF1EN | RW | VTF interrupt 1 enable bit.<br>1: VTF interrupt 1 channel is enabled;<br>0: Off. | 0 |

### 5.5.2.10 PFIC VTF Interrupt 2 Address Register (PFIC_VTFADDRR2)

Offset address: 0x68

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR2[31:16] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR2[15:1] | | | | | | | | | | | | | | | VTF2EN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:1] | ADDR2 | RW | VTF interrupt 2 service program address bit[31:1]. | 0 |
| 0 | VTF2EN | RW | VTF interrupt 2 enable bit.<br>1: VTF interrupt 2 channel is enabled;<br>0: Off. | 0 |

### 5.5.2.11 PFIC VTF Interrupt 3 Address Register (PFIC_VTFADDRR3)

Offset address: 0x6C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR3[31:16] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR3[15:1] | | | | | | | | | | | | | | | VTF3E |

| | |
|---|---|
| | N |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:1] | ADDR3 | RW | VTF interrupt 3 service program address bit[31:1]. | X |
| 0 | VTF3EN | RW | VTF interrupt 3 enable bit.<br>1: VTF interrupt 3 channel is enabled;<br>0: Off. | 0 |

### 5.5.2.12 PFIC Interrupt Enable Setting Register 1 (PFIC_IENR1)

Offset address: 0x100

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTEN[31:16] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTEN15 | INTEN14 | INTEN13 | INTEN12 | Reserved | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | INTEN | WO | 12#-31# interrupt enable control.<br>1: Current number interrupt enable.<br>0: No effect. | 0 |
| [11:0] | Reserved | RO | Reserved. | 0 |

### 5.5.2.13 PFIC Interrupt Enable Setting Register 2 (PFIC_IENR2)

Offset address: 0x104

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | INTEN[46:32] | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved | 0 |
| [14:0] | INTEN | WO | 32#-46# interrupt enable control.<br>1: Current number interrupt enable.<br>0: No effect. | 0 |

### 5.5.2.14 PFIC Interrupt Enable Clear Register 1 (PFIC_IRER1)

Offset address: 0x180

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTRSET[31:16] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTRSET15 | INTRSET14 | INTRSET13 | INTRSET12 | Reserved | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | INTRSET | WO | 12#-31# interrupt disable control.<br>1: Current number interrupt disable.<br>0: No effect. | 0 |
| [11:0] | Reserved | RO | Reserved. | 0 |

### 5.5.2.15 PFIC Interrupt Enable Clear Register 2 (PFIC_IRER2)

Offset address: 0x184

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | INTRSET[46:32] | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved | 0 |
| [14:0] | INTRSET | WO | 32#-46# interrupt disable control.<br>1: Current number interrupt disable.<br>0: No effect. | 0 |

### 5.5.2.16 PFIC Interrupt Pending Setting Register 1 (PFIC_IPSR1)

Offset address: 0x200

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PENDSET[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PENDSET15 | PENDSET14 | PENDSET13 | PENDSET12 | Reserved | | PENDSET9 | PENDSET8 | Reserved | | PENDSET5 | Reserved | PENDSET3 | PENDSET2 | Reserved | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | PENDSET | WO | 12#-31# interrupt pending settings:<br>1: Current number interrupt pending;<br>0: No effect. | 0 |
| [11:10] | Reserved | RO | Reserved | 0 |
| [9:8] | PENDSET | WO | 8#-9# interrupt pending settings:<br>1: Current number interrupt pending;<br>0: No effect. | 0 |
| [7:6] | Reserved | RO | Reserved | 0 |
| 5 | PENDSET | WO | 5# interrupt pending settings:<br>1: Current number interrupt pending;<br>0: No effect. | 0 |
| 4 | Reserved | RO | Reserved | 0 |
| [3:2] | PENDSET | WO | 2#-3# interrupt pending settings:<br>1: Current number interrupt pending;<br>0: No effect. | 0 |
| [1:0] | Reserved | RO | Reserved | 0 |

### 5.5.2.17 PFIC Interrupt Pending Setting Register 2 (PFIC_IPSR2)

Offset address: 0x204

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | PENDSET[46:32] | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved | 0 |
| [14:0] | PENDSET | WO | 32#-46# interrupt pending settings:<br>1: Current number interrupt pending;<br>0: No effect. | 0 |

### 5.5.2.18 PFIC Interrupt Pending Clear Register 1 (PFIC_IPRR1)

Offset address: 0x280

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PENDRST[31:16] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PENDRST15 | PENDRST14 | PENDRST13 | PENDRST12 | Reserved | | PENDRST9 | PENDRST8 | Reserved | | PENDRST5 | Reserved | PENDRST3 | PENDRST2 | Reserved | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | PENDRST | WO | 12#-31# interrupt pending cleared:<br>1: Current numbered interrupt clears the pending status;<br>0: No effect. | 0 |
| [11:10] | Reserved | RO | Reserved | 0 |
| [9:8] | PENDRST | WO | 8#-9# interrupt pending cleared:<br>1: Current numbered interrupt clears the pending status;<br>0: No effect. | 0 |
| [7:6] | Reserved | RO | Reserved | 0 |
| 5 | PENDRST | WO | 5# interrupt pending cleared:<br>1: Current numbered interrupt clears the pending status;<br>0: No effect. | 0 |
| 4 | Reserved | RO | Reserved | 0 |
| [3:2] | PENDRST | WO | 2#-3# interrupt pending cleared:<br>1: Current numbered interrupt clears the pending status;<br>0: No effect. | 0 |
| [1:0] | Reserved | RO | Reserved | 0 |

### 5.5.2.19 PFIC Interrupt Pending Clear Register 2 (PFIC_IPRR2)

Offset address: 0x284

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| Reserved | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | PENDRST[46:32] | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved | 0 |
| [14:0] | PENDRST | WO | 32#-46# interrupt pending cleared:<br>1: Current numbered interrupt clears the pending status;<br>0: No effect. | 0 |

### 5.5.2.20 PFIC Interrupt Activation Status Register 1 (PFIC_IACTR1)

Offset address: 0x300

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IACTS[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IACTS15 | IACTS14 | IACTS13 | IACTS12 | Reserved | | IACTS9 | IACTS8 | Reserved | | IACTS5 | Reserved | IACTS3 | IACTS2 | Reserved | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | IACTS | RO | 12#-31# interrupt execution status.<br>1: Current number interrupt in execution.<br>0: Current number interrupt is not executed. | 0 |
| 15 | Reserved | RO | Reserved. | 0 |
| [9:8] | IACTS | RO | 8#-9# interrupt execution status.<br>1: Current number interrupt in execution.<br>0: Current number interrupt is not executed. | 0 |
| [7:6] | Reserved | RO | Reserved. | 0 |
| 5 | IACTS | RO | 5# interrupt execution status.<br>1: Current number interrupt in execution.<br>0: Current number interrupt is not executed. | 0 |
| 4 | Reserved | RO | Reserved. | 0 |
| [3:2] | IACTS | RO | 2#-3# interrupt execution status.<br>1: Current number interrupt in execution.<br>0: Current number interrupt is not executed. | 0 |
| [1:0] | Reserved | RO | Reserved. | 0 |

### 5.5.2.21 PFIC Interrupt Activation Status Register 2 (PFIC_IACTR2)

Offset address: 0x304

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | IACTS[46:32] | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved. | 0 |
| [14:0] | IACTS | RO | 32#-46# interrupt execution status.<br>1: Current number interrupt in execution.<br>0: Current number interrupt is not executed. | 0 |

### 5.5.2.22 PFIC Interrupt Priority Configuration Register (PFIC_IPRIORx) (x=0-63)

Offset address: 0x400-0x4FF

The controller supports 256 interrupts (0-255), each using 8 bits to set the control priority.

| | 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|---|
| IPRIOR63 | PRIO_255 | PRIO_254 | PRIO_253 | PRIO_252 |
| … | … | … | … | … |
| IPRIORx | PRIO_(4x+3) | PRIO_(4x+2) | PRIO_(4x+1) | PRIO_(4x) |
| … | … | … | … | … |
| IPRIOR0 | PRIO_3 | PRIO_2 | PRIO_1 | PRIO_0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [2047:2040] | IP_255 | RW | Same as IP_0 description. | 0 |
| … | … | … | … | … |
| [31:24] | IP_3 | RW | Same as IP_0 description. | 0 |
| [23:16] | IP_2 | RW | Same as IP_0 description. | 0 |
| [15:8] | IP_1 | RW | Same as IP_0 description. | 0 |
| [7:0] | IP_0[7:0] | RW | Number 0 interrupt priority configuration.<br>[7:5]: Priority control bits.<br>If no nesting is configured, no preemption bits;<br>If configured with 2 levels of nesting, bit 7 is the preemption bit;<br>The smaller the priority value, the higher the priority. If interrupts of the same preemption priority hang at the same time, the interrupt with the higher priority is executed first.<br>[4:0]: Reserved, fixed to 0, write invalid. | 0 |

*Note: The priority of interrupts 2, 3, 5, 8 and 9 is fixed at 0, and cannot be configured.*

### 5.5.2.23 PFIC System Control Register (PFIC_SCTLR)

Offset address: 0xD10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SYSRESET | Reserved | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | SETEVENT | SEVONPEND | WFITOWFE | SLEEPDEEP | SLEEPONEXIT | Reserved |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31 | SYSRESET | WO | System reset, clear 0 automatically. write 1 | 0 |

| | | | | |
|---|---|---|---|---|
| | | | valid, write 0 invalid. | |
| [30:6] | Reserved | RO | Reserved. | 0 |
| 5 | SETEVENT | WO | Set the event to wake up the WFE case. | 0 |
| 4 | SEVONPEND | RW | When an event occurs or interrupts a pending state, the system can be woken up from after the WFE instruction, or if the WFE instruction is not executed, the system will be woken up immediately after the next execution of the instruction.<br>1: enabled events and all interrupts (Including unenabled interrupts) can wake up the system.<br>0: Only enabled events and enabled interrupts can wake up the system. | 0 |
| 3 | WFITOWFE | RW | Execute the WFI command as if it were a WFE.<br>1: treat the subsequent WFI instruction as a WFE instruction.<br>0: No effect. | 0 |
| 2 | SLEEPDEEP | RW | Low-power mode of the control system.<br>1：deepsleep 0：sleep | 0 |
| 1 | SLEEPONEXIT | RW | System status after control leaves the interrupt service program.<br>1: The system enters low-power mode.<br>0: The system enters the main program. | 0 |
| 0 | Reserved | RO | Reserved. | 0 |

### 5.5.3 STK Register Description

Table 5-5 STK-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_STK_CTLR | 0xE000F000 | System count control register | 0x00000000 |
| R32_STK_SR | 0xE000F004 | System count status register | 0x00000000 |
| R32_STK_CNTL | 0xE000F008 | System counter low register | 0x00000000 |
| R32_STK_CMPLR | 0xE000F010 | Counting comparison low register | 0x00000000 |

#### 5.5.3.1 System Count Control Register (STK_CTLR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | | MODE | STRE | STCLK | STIE | STE |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:5] | Reserved | RO | Reserved. | 0 |
| 4 | MODE | RW | Counting modes:<br>1: Counting down;<br>0: Counting up. | 0 |
| 3 | STRE | RW | Auto-reload count enable bit.<br>1: Re-counting from 0 after counting up to the comparison value.<br>0: Count up to the comparison value and continue | 0 |

| | | | | |
|---|---|---|---|---|
| | | | counting up, count down to 0 and start counting down again from the maximum value. | |
| 2 | STCLK | RW | Counter clock source selection bit.<br>1: HCLK for time base.<br>0: HCLK/8 for time base. | 0 |
| 1 | STIE | RW | Counter interrupt enable control bit.<br>1: Enable counter interrupt.<br>0: Disable counter interrupt. | 0 |
| 0 | STE | RW | System counter enable control bit.<br>1: Turn on the system counter STK.<br>0: Turn off the system counter STK and the counter stops counting. | 0 |

### 5.5.3.2 System Count Status Register (STK_SR)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWIE | Reserved | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | | | | | | CNTIF |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31 | SWIE | RW | Software interrupt trigger enable (SWI):<br>1: Trigger software interrupt;<br>0: Turn off the trigger.<br>*Note: This bit must be cleared after entering the software interrupt, otherwise it will always trigger.* | 0 |
| [30:1] | Reserved | RO | Reserved | 0 |
| 0 | CNTIF | RW0 | Count value comparison flag, write 0 to clear, write 1 to invalidate.<br>1: Counting up to the comparison value and down to 0;<br>0: The comparison value is not reached. | 0 |

### 5.5.3.3 System Counter Low Register (STK_CNTL)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CNT[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNT[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | CNT[31:0] | RW | The current counter count value is 32 bits lower. | X |

### 5.5.3.5 Counting Comparison Low Register (STK_CMPLR)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMP[31:16] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMP[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | CMP[31:0] | RW | Sets the comparison counter value 32 bits lower. | 0 |

### 5.5.4 Dedicated CSR Registers

A number of Control and Status Registers (CSRs) are defined in the RISC-V architecture to configure or identify or record operational status. In addition to the standard registers defined in the RISC-V Privileged Architecture document, the CH32M030 chip also has a number of vendor-defined registers that need to be accessed using the csr instruction.

*Note: Some CSR registers need to be accessed by the system in machine mode, and accessing these CSR registers in non-machine mode will cause the chip to enter an exception. Marked as "MRW": the system needs to be in machine mode to access; Marked as "MRO": read-only in machine mode.*

Table 5-6 List of CSR Related Registers

| Name | CSR address | Access | Deacription | Reset value |
|---|---|---|---|---|
| MTVEC | 0x305 | MRW | Machine mode exception base address register | 0x00000000 |
| GINTENR | 0x800 | URW | User mode global interrupt enable register | 0x00000000 |
| INTSYSCR | 0x804 | URW | Interrupt system control register | 0x0000E002 |
| CORECFGR | 0xBC0 | MRW | Microprocessor configuration register | 0x00000001 |
| INESTCR | 0xBC1 | MRW | Interrupt nested control register | 0x00000000 |

#### 5.5.4.1 Machine Interrupt Vector Base Address Register (MTVEC)

CSR address: 0x305

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:2] | BASEADDR[31:2] | MRW | Interrupt vector table base address, where bits [9:2] are fixed to 0. | 0 |
| 1 | MODE1 | MRW | Interrupt vector table identification mode:<br>1. Identify by absolute address, support full range, but must jump;<br>0: Identify by jump instruction, with limited range, and support non-jump instruction. | 0 |
| 0 | MODE0 | MRW | Interrupt or abnormal entry address mode selection:<br>1: Address offset according to interrupt number *4;<br>0: Use uniform entry address. | 0 |

For MCU of V3 series microprocessor, MODE0 is configured as 1 by default in the startup file, and the entry of exception or interrupt is shifted according to the interrupt number *4. Note that V3B microprocessor can be either a jump instruction or an absolute address of the interrupt function, and it is configured as an absolute address by default in the startup file.

#### 5.5.4.2 User mode global interrupt enable register (GINTENR)

This register is used to control the enable and mask of global interrupt. The enable and mask of global interrupt in machine mode can be controlled by the MIE and MPIE bits in mstatus, but this register cannot be operated in user mode. The global interrupt enable register gintenr is the mapping of MIE and MPIE in mstatus, and can be used to

set and clear MIE and MPIE by operating gintenr in user mode.

CSR address: 0x800

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:13] | Reserved | URO | Reserved | 0 |
| [12:11] | MPP[1:0] | URO | Enter privileged mode before interruption. | 0 |
| [10:8] | Reserved | URO | Reserved | 0 |
| 7 | MPIE | URW | When 0xBC0(CSR)bit5 is enabled, this bit can be read and written in user mode. | 0 |
| [6:4] | Reserved | URO | Reserved | 0 |
| 3 | MIE | URW | When 0xBC0(CSR)bit5 is enabled, this bit can be read and written in user mode. | 0 |
| [1:0] | Reserved | URO | Reserved | 0 |

### 5.5.4.3 Interrupt System Control Register (INTSYSCR)

CSR address: 0x804

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:6] | Reserved | URO | Reserved. | 0x380 |
| 5 | GIHWSTKNEN | URW1 | Global interrupt and hardware stack off enable.<br>*Note: This bit is often used in real time operating systems, when the interrupt switches context, set this bit to turn off the global interrupt and hardware pressure stack out of the stack, when the context switch is completed, after the execution of the interrupt return, the hardware automatically clears this bit.* | 0 |
| [4:2] | Reserved | URO | Reserved. | 0 |
| 1 | INESTEN | URO | The interrupt nesting function is enabled, and the fixed value is 1:<br>1: Enable;<br>0: Disable.<br>*Note: The actual nesting level is controlled by NEST_LVL in CSR 0xBC1.* | 0 |
| 0 | HWSTKEN | URW | Hardware stack enable:<br>1: Hardware stacking function enabled;<br>0: Hardware stacking function disabled. | 0 |

### 5.5.4.4 Core Configuration Register (CORECFGR)

CSR address: 0XBC0

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:8] | Reserved | MRO | Reserved | 0 |
| 7 | CSTA_FAULT_IE | MRW | Core state error interrupt enable:<br>1. When a state error occurs, an NMI interrupt is generated;<br>0: When a status error occurs, no NMI interrupt is generated. | 0 |
| 6 | Reserved | MRO | Reserved, keep 0. | 0 |
| 5 | IE_REMAP_EN | MRW | MIE register mapping enable:<br>1: Bits 3 and 7 of CSR address 0x800 are mapped to bit MIE of the MSTATUS register and bit MPIE of the MSTATUS register, respectively;<br>0: CSR address 0x800 is a read-only register, and the value returned is MSTATUS. | 0 |
| [4:2] | Reserved | MRO | Reserved | 0 |

| | | | | |
|---|---|---|---|---|
| [1:0] | FETCH_MODE[1:0] | MRW | Fetching mode:<br>00: Prefetch is off. The instruction prefetch function is turned off to avoid invalid instruction fetching operation, and there is at most one valid instruction on the CPU pipeline. This mode has the lowest power consumption, and its performance drops by about 2 ~ 3 times.<br>01: Prefetch mode 1. When the instruction prefetch function is turned on, the CPU will continue to access the instruction memory until the number of instructions to be executed in the internal instruction buffer exceeds a certain number, or the instruction buffer is full, and instruction fetching will be suspended; (Failure of CPU prediction will lead to redundant fetch operation, and in some cases, the execution unit will introduce 0 ~ 2 cycles of bubbles, and the performance of most programs will not decrease obviously);<br>Other: Reserved. | 0x1 |

### 5.5.4.5 Core Configuration Register (INESTCR)

CSR address: 0XBC1

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31 | Reserved | MRO | Reserved | 0 |
| 30 | NEST_OV | MRW | Interrupt/exception nested overflow flag bit, write 1 to clear:<br>1: Interrupt overflow flag;<br>0: Interrupt did not overflow.<br>*Note: Interrupt overflow will only occur when executing secondary interrupt service function to generate instruction exception or NMI interrupt. At this time, the exception and NMI interrupt enter normally, but the CPU stack overflows, so you cannot exit from this exception and NMI interrupt.* | 0 |
| [29:12] | Reserved | MRO | Reserved | 0 |
| [11:8] | NEST_STA[3:0] | MRO | Nested status flag bit:<br>0000: No interrupt;<br>0001: 1-level interrupt;<br>0011: 2-level interrupt (1-level nesting);<br>0111: 3-level interrupt (Overflow);<br>1111: 4-level interrupt (Overflow). | 0 |
| [7:2] | Reserved | MRO | Reserved | 0 |
| [1:0] | NEST_LVL[1:0] | MRW | Nesting level:<br>00: Nesting is prohibited and the nesting function is turned off;<br>01: 1-level nesting, which turns on the nesting function;<br>Other: invalid.<br>*Note: Write 10 or 11 to this field, and the field will be set to 01. When write 11 to this field, read this register to get the highest nesting level of the chip.* | 0 |

# Chapter 6 GPIO and Alternate Function (GPIO/AFIO)

The system provides 3 groups of GPIO ports, all GPIO pins can be configured as outputs by software, and most GPIO pins can be configured as input or multiplexed peripheral functional ports. Most GPIO pins are shared with digital or analog alternate peripherals, providing a locking mechanism to freeze the IO configuration to avoid accidental writing to I/O registers.

PB9, PB11, PB13 and PB15 only support push-pull output, and the default output is low, and input is not supported.

PB7 only supports high-voltage open-drain output, not input; PC5 supports push-pull output under light load, and can prevent current from flowing backwards when the voltage difference is below 7V.

All GPIO pins support controllable pull-up except PB7~PB15 and PC5. Among them, PB0 and PB1 have built-in pull-up resistors that can be turned off by default, adjusted and turned on, which are adjusted and controlled by two groups of PDE and DAC in EXTEN_CTLR1, and can provide pull-up current; CC1/CC2/CC3/CC4 provide multistage pull-up current.

PB0 and PB1 have built-in pull-down resistors that can be turned off by default, adjusted and turned on, which are adjusted and controlled by two groups of PDE and DAC in EXTEN_CTLR1, and can provide pull-down current; CC1/CC2/CC3/CC4 if there is a suffix r, it means that the controllable Rd pull-down resistor defined by the built-in Type-C specification is turned on by default; CCxR pin has a controllable Rd pull-down resistor defined by Type-C specification, so it is recommended to turn off the pull-down when it is used as a GPIO push-pull output. PB8, PB10, PB12 and PB14 have built-in pull-down resistors that cannot be turned off; None of the other GPIO pins have built-in pull-down resistors.

## 6.1 Main Features

Each pin of the port can be configured to one of the following multiple modes.

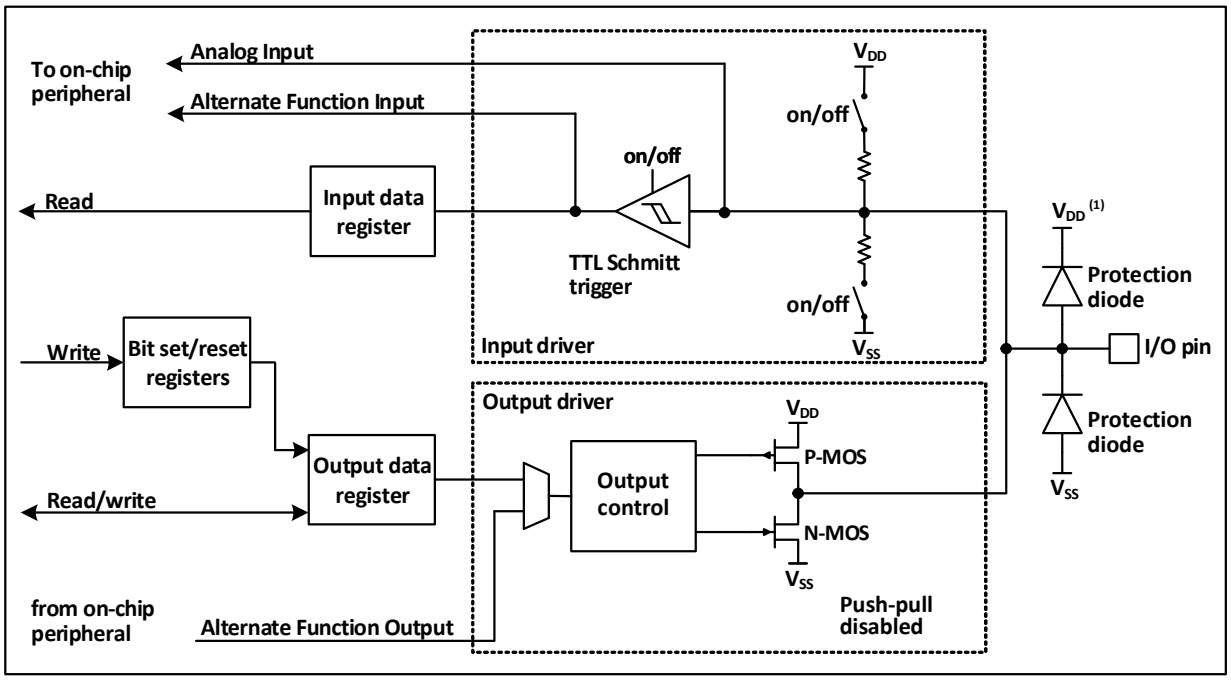
- Floating input
- Pull-up input
- Pull-down input (partial IO)
- Analog input
- Push-pull output
- Open-drain input
- Inputs and outputs of alternate functions

Many pins have alternate capabilities, and many other peripherals map their output and input channels to these pins. The specific usage of these alternate pins needs to be referred to the individual peripherals, and the content of whether these pins are alternate and remapped is explained in this chapter.

# 6.2 Function Description

## 6.2.1 Overview

Figure 6-1 GPIO module basic structure block diagram




As shown in Figure 8-1 IO port structure, each pin has two protection diodes inside the chip, and the IO port can be internally divided into input and output driver modules. The input driver has a pull-up resistor and a pull-down resistor (Only for some of the IOs) to choose from, and can be connected to peripherals with analogue inputs such as AD; if the input is to a digital peripheral, it needs to go through a TTL Schmitt trigger and then be connected to a GPIO input register or other multiplexed peripherals. The output driver, on the other hand, has a pair of MOS tubes that can configure the IO port as a push-pull output; the output driver can also be configured internally as to whether the output is controlled by the GPIO or by another peripheral that is multiplexed.

## 6.2.2 GPIO Initialization Function

Just after a reset, the GPIO ports are running in their initial state, when most IO ports are running in the floating input state, but there are also peripheral related pins that are running on the peripheral multiplexing function. Please refer to the relevant section of the pin description for the specific initialization function.

## 6.2.3 External Interrupts

All GPIO ports can be configured with external interrupt input channels, but an external interrupt input channel can only be mapped to at most one GPIO pin, and the serial number of the external interrupt channel must be the same as the bit number of the GPIO port, for example, PA1 (Or PB1, PC1) can only be mapped to EXTI1, and EXTI1 can only accept one of PA1, PB1 or PC1, etc. The mapping of both parties is one-to-one.

## 6.2.4 Alternate Functions

It is important to note that using the alternate function.

- To use the alternate function in the input direction, the port must be configured in alternate input mode, and

the pull-down settings can be set according to actual needs.

- To use the alternate function in the output direction, the port must be configured in alternate output mode.
- For bidirectional alternate function, the port must be configured in alternate output mode, when the driver is configured in floating input mode

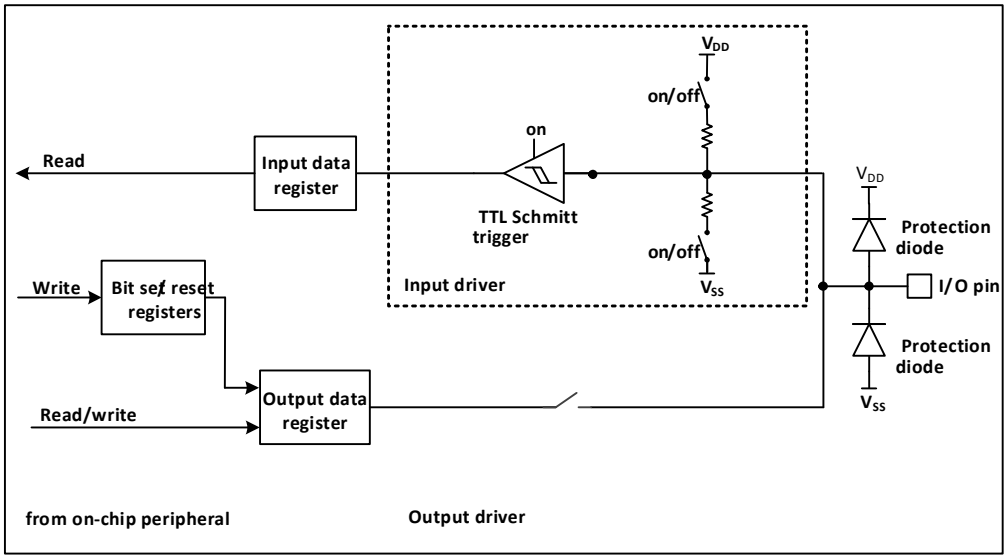
The same I/O port may have multiple peripherals alternate to this pin, so in order to maximize the space for each peripheral, the alternate pins of peripherals can be remapped to other pins in addition to the default alternate pins, avoiding the occupied pins.

### 6.2.5 Locking Mechanism

The locking mechanism locks the configuration of the I/O port. After a specific write sequence, the selected I/O pin configuration will be locked and cannot be changed until the next reset.

### 6.2.6 Input Configuration

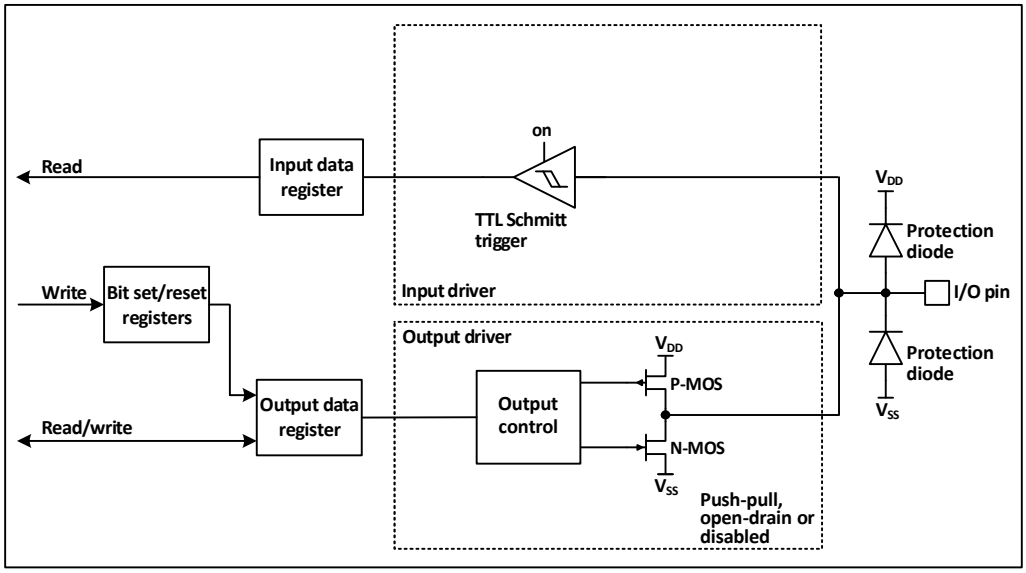
Figure 6-2 GPIO module input configuration structure block diagram




When the IO port is configured in input mode, the output driver is disconnected, and the input pull-up and pull-down are optional (only some IO supports controllable pull-down), and the multiplexing function and analog input are not connected. The data on each IO port will be sampled into the input data register at each HB clock, and the level state of the corresponding pin will be obtained by reading the corresponding bit of the input data register.

### 6.2.7 Output Configuration

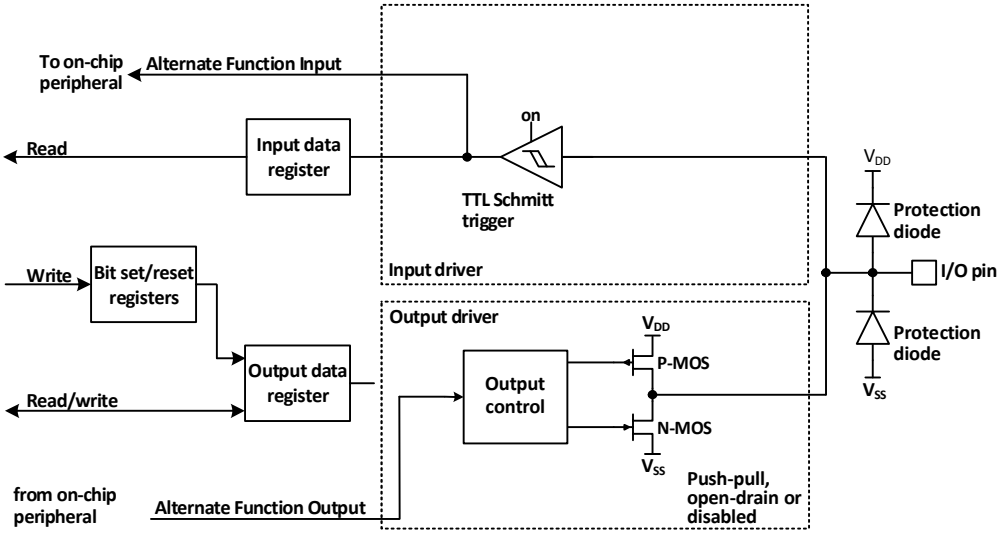
Figure 6-3 GPIO module output configuration structure block diagram




When the IO port is configured in the output mode, a pair of MOS in the output driver can be configured in the push-pull or open-drain mode as required without using the alternate function. Input-driven pull-up and pull-down resistors are disabled, TTL Schmitt trigger is activated, and the level appearing on the IO pin will be sampled into the input data register at every HB clock, so reading the input data register will get the IO state, and in the push-pull output mode, accessing the output data register will get the last written value.

### 6.2.8 Alternate Function Configuration

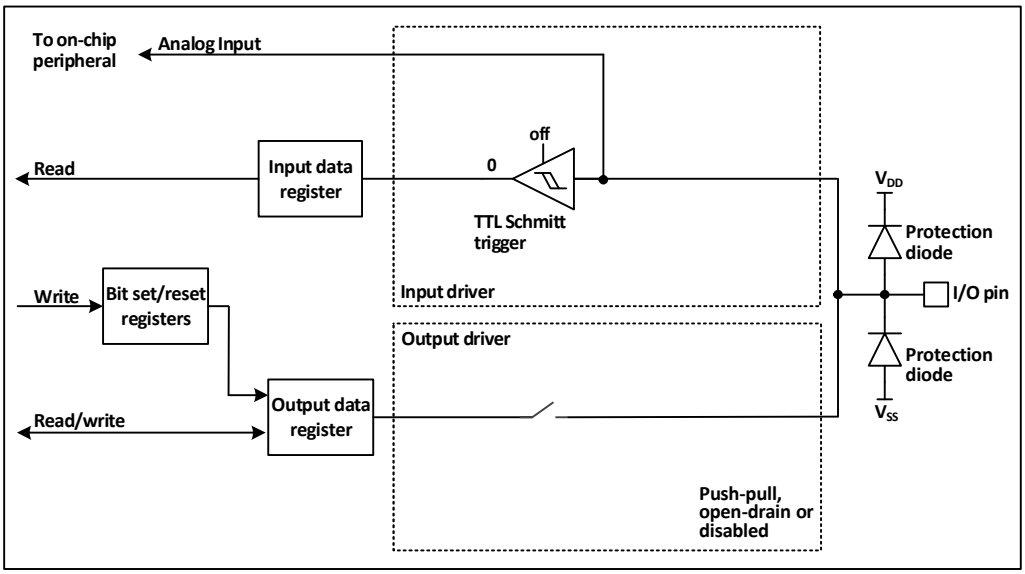
Figure 6-4 The structure of GPIO module when it is alternate by other peripherals




When the alternate function is enabled, the output driver is enabled and can be configured in open-drain or push-pull mode as required, the Schmidt trigger is also turned on, the input and output lines of the multiplexing function are connected, but the output data register is disconnected, and the level appearing on the IO pin will be sampled into the input data register every HB clock. In the open-drain mode, reading the input data register will get the current state of the IO port. In push-pull mode, reading the output data register will get the last written value.

### 6.2.9 Analog Input Configuration

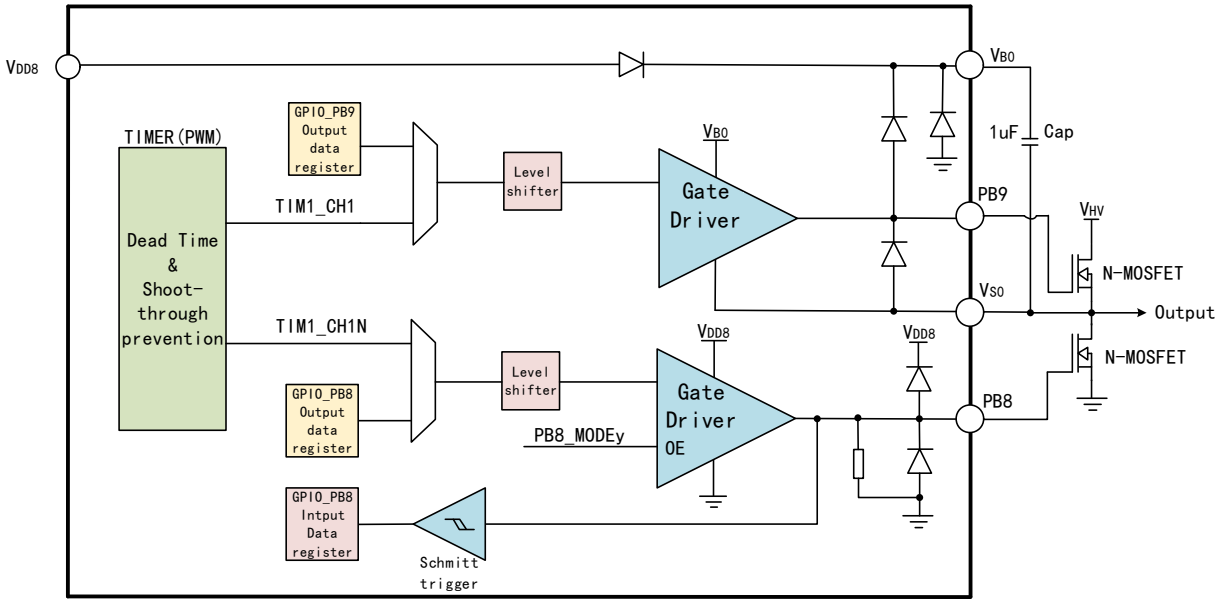
Figure 6-5 The configuration structure when the GPIO module is used as an analog input




When the analog input is enabled, the output buffer is disconnected, the input of the Schmitt trigger in the input driver is disabled to prevent the generation of consumption on the I/O port, the pull-up and pull-down resistors are disabled, and the read input data register will always be 0.

### 6.2.10 GPIO structure of MV

Figure 6-6 MV GPIO structure block diagram




*Note: Cap is an off-chip capacitor of 1uF; N-MOSFET is an off-chip N-MOSFET power transistor.*

CH32M030 integrates 4 independent half-bridge drivers, each half-bridge includes a low-voltage bootstrap diode, high-side and low-side level shift circuits, high-side and low-side output drive circuits, and supports four pairs of gate drives of N-type MOSFET power tubes. Only one capacitor is needed to store the bootstrap power supply

externally, and the gate drive voltage depends on VDD8, and it can be adjusted from 5V to 10V in four steps.

PB9 and PB8 are respectively controlled as high-side and low-side power tubes driven by the first half bridge, PB11 and PB10 are respectively controlled as high-side and low-side power tubes driven by the second half bridge, PB13 and PB12 are respectively controlled as high-side and low-side power tubes driven by the third half bridge, and PB15 and PB14 are respectively controlled as high-side and low-side power tubes driven by the fourth half bridge.

4 independent half-bridges can form a three-phase half-bridge, which is used to drive the grid of a three-phase motor and is controlled by PB8~PB13 signals generated by TIM1. Four independent half-bridges can also form two groups of full-bridges for two independent full-bridge drives, which are controlled by PB8~PB11 signals generated by TIM1 and PB12~PB15 signals generated by TIM2 respectively. The timer generates PWM signal to support dead time control.

In addition to half-bridge gate drive, PB8~PB15 can also be used for GPIO. The input supports $V_{DD8}$ withstand voltage, and the output swings with reference to $V_{DD8}$ voltage. Among them, when PB9, PB11, PB13 and PB15 are used for GPIO, it is necessary to connect $V_S$ to GND as a GPIO powered by $V_{DD8}$; You can also connect $V_S$ and $V_B$ to the low-end and high-end of the floating voltage respectively, and the voltage is $V_B$ and $V_S$ when the high/low level is output.

### 6.2.11 GPIO Settings for Peripherals

The following table recommends the corresponding GPIO port configuration for each peripheral pin.

Table 6-1 Advanced-control timer (TIM1)

| TIM1/2 | Configuration | GPIO configuration |
|---|---|---|
| TIM1_CHx | Input capture channel x | Floating input |
| | Output compare channel x | Push-pull alternate output |
| TIM1_CHxN | Complementary output channels x | Push-pull alternate output |
| TIM1_BKIN | Brake input | Floating input |
| TIM1_ETR | Externally triggered clock input | Floating input |

Table 6-2 General-purpose timer (TIM2)

| TIM2 pin | Configuration | GPIO configuration |
|---|---|---|
| TIM2_CHx | Input capture channel x | Floating input |
| | Output comparison channel x | Push-pull alternate output |
| TIM2_ETR | Externally triggered clock input | Floating input |

Table 6-3 General-purpose timer (TIM3)

| TIM3 pin | Configuration | GPIO configuration |
|---|---|---|
| TIM3_CHx | Input capture channel x | Floating input |
| | Output comparison channel x | Push-pull alternate output |
| TIM3_CHxN | Complementary output channel x | Push-pull alternate output |

Table 6-4 Universal asynchronous serial transceiver (UART)

| USART pin | Configuration | GPIO configuration |
|---|---|---|
| UART1_TX | Full-duplex mode | Push-pull alternate output |
| | Half-duplex synchronous mode | Open-drain alternate output |
| UART1_RX | Full-duplex mode | Floating input or pull-up input |

| | | |
|---|---|---|
| | Half-duplex synchronous mode | Not used |
| UART1_RTS | Hardware flow control | Push-pull alternate output |
| UART1_CTS | Hardware flow control | Floating input or pull-up input |

Table 6-5 Serial peripheral interface (SPI) modules

| SPI pin | Configuration | GPIO configuration |
|---|---|---|
| SPI1_SCK | Master mode | Push-pull alternate output |
| | Slave mode | Floating input |
| SPI1_MOSI | Full-duplex Master mode | Push-pull alternate output |
| | Full-duplex Slave mode | Floating input or pull-up input |
| | Simple bi-directional data line/Master mode | Push-pull alternate output |
| | Simple bi-directional data line/Slave mode | Not used |
| SPI1_MISO | Full-duplex Master mode | Floating input or pull-up input |
| | Full-duplex Slave mode | Push-pull alternate output |
| | Simple bi-directional data line/Master mode | Not used |
| | Simple bi-directional data line/Slave mode | Push-pull alternate output |
| SPI1_NSS | Hardware master/slave mode | Floating, pull-up input |
| | Hardware master mode/NSS output enable | Push-pull alternate output |
| | Software mode | Not used |

Table 6-6 Internal integrated bus (I2C1) module

| I2C pin | Configuration | GPIO configuration |
|---|---|---|
| I2C1_SCL | I2C clock | Open-drain alternate output |
| I2C1_SDA | I2C data | Open-drain alternate output |

Table 6-7 Analog-to-digital converters (ADC)

| ADC pin | GPIO configuration |
|---|---|
| ADC | Analog input |

Table 6-8 Other I/O function settings

| Pin | Configuration function | GPIO configuration |
|---|---|---|
| MCO | Clock output | Push-pull alternate output |
| EXTI | External interrupt input | Floating, pull-up input |
| OPA | Operational amplifier input | Floating input |

## 6.2.12 GPIO Settings for Peripherals

### 6.2.12.1 Timer Alternate Function Remapping

Table 6-9 TIM1 alternate function remapping

| Alternate function | TIM1_RM=000 Default mapping | TIM1_RM=001 Partial mapping | TIM1_RM=010 Partial mapping | TIM1_RM=011 Partial mapping | TIM1_RM=100 Partial mapping |
|---|---|---|---|---|---|
| TIM1_ETR | PC1 | PC1 | PA4 | PA4 | PA4 |
| TIM1_CH1 | PB9 | PB9 | PB9 | PC3 | PC3/PB9 |
| TIM1_CH2 | PB11 | PB11 | PB11 | PC4 | PC4/PB11 |
| TIM1_CH3 | PB13 | PB13 | PB15 | PA0 | PA0/PB13 |

| | | | | | |
|---|---|---|---|---|---|
| TIM1_CH4 | PC0 | PB15 | PC2 | PC5 | PA1 |
| TIM1_CH1N | PB8 | PB8 | PB8 | PC2 | PC2/PB8 |
| TIM1_CH2N | PB10 | PB10 | PB10 | PC1 | PC1/PB10 |
| TIM1_CH3N | PB12 | PB12 | PB14 | PC0 | PC0/PB12 |
| TIM1_BKIN | PA15 | PA13 | PA15 | PA15 | PA15 |

*Note: When RMP=100, CH1/CH1N, CH2/CH2N and CH3/CH3N of TIM1 output the same waveform from 2 different pads respectively.*

Table 6-10 TIM2 alternate function remapping

| Alternate function | TIM2_RM=00 Default mapping | TIM2_RM=01 Partial mapping | TIM2_RM=10 Full mapping | TIM2_RM=11 Full mapping |
|---|---|---|---|---|
| TIM2_ETR | PA5 | PA3 | PB13 | PB5 |
| TIM2_CH1 | PA5 | PA3 | PB13 | PB5 |
| TIM2_CH2 | PA6 | PB0 | PB15 | PB6 |
| TIM2_CH3/TIM2_CH1N | PA7 | PB1 | PB12 | PB1 |
| TIM2_CH4/TIM2_CH2N | PA4 | PB2 | PB14 | PB2 |

Table 6-11 TIM3 alternate function remapping

| Alternate function | TIM3_RM=000 Default mapping | TIM3_RM=001 Partial mapping | TIM3_RM=010 Full mapping | TIM3_RM=011 Partial mapping | TIM3_RM=100 Full mapping |
|---|---|---|---|---|---|
| TIM3_CH1 | PC0 | PC0 | PB2 | PA2 | PB13 |
| TIM3_CH2 | PC1 | PB2 | PB3 | PB5 | PB15 |
| TIM3_CH1N | PB2 | PC1 | PB4 | PA4 | PB12 |
| TIM3_CH2N | PB3 | PB3 | PB5 | PB6 | PB14 |

### 6.2.12.2 UART Alternate Function Remapping

Table 6-12 UART1 alternate function remapping

| Alternate function | UART1_RM=000 Default mapping | UART1_RM=001 Remapping | UART1_RM=010 Remapping | UART1_RM=011 Remapping | UART1_RM=100 Remapping | UART1_RM=101 Remapping |
|---|---|---|---|---|---|---|
| UART1_TX | PC1 | PC1 | PB6 | PB1 | PA3 | PA2 |
| UART1_RX | PC0 | PC2 | PB5 | PB0 | PA2 | PA3 |
| UART1_CTS | PA14 | PA14 | PA2 | PB2 | PB2 | PB2 |
| UART1_RTS | PA15 | PA15 | PA4 | PB3 | PB3 | PB3 |

### 6.2.12.3 SPI Alternate Function Remapping

Table 6-13 SPI alternate function remapping

| Alternate function | SPI_RM=00 Default mapping | SPI_RM=01 Remapping | SPI_RM=10 Remapping | SPI_RM=11 Remapping |
|---|---|---|---|---|
| SPI_NSS | PA0 | PA2 | PA0 | PA2 |
| SPI_SCK | PA1 | PB6 | PA1 | PA1 |

| | | | | |
|---|---|---|---|---|
| SPI_MISO | PC4 | PA4 | PC4 | PA4 |
| SPI_MOSI | PC3 | PB5 | PA11 | PA8 |

#### 6.2.12.4 I2C Alternate Function Remapping

Table 6-14 I2C alternate function remapping

| Alternate function | I2C1_RM=00 Default mapping | I2C1_RM=01 Remapping | I2C1_RM=10 Remapping | I2C1_RM=11 Remapping |
|---|---|---|---|---|
| I2C1_SCL | PB3 | PA2 | PA15 | PA2 |
| I2C1_SDA | PB2 | PB6 | PA14 | PA3 |

#### 6.2.12.5 ADC Alternate Function Remapping

Table 6-15 ADC External trigger conversion alternate function remapping

| Alternate function | ADC_ETRGIN_RM=0 Default mapping | ADC_ETRGIN_RM=1 Remapping |
|---|---|---|
| ADC External trigger conversion | ADC external trigger conversion connected to PB6 | ADC external trigger conversion connected to PA14 |

## 6.3 Register Description

### 6.3.1 GPIO Register Description

Unless otherwise specified, the registers of the GPIO must be operated as words (Operate these registers with 32 bits).

Table 6-16 GPIO-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_GPIOA_CFGLR | 0x40010800 | PA port configuration register low | 0x44444444 |
| R32_GPIOA_CFGHR | 0x40010804 | PA port configuration register high | 0x44444444 |
| R32_GPIOB_CFGLR | 0x40010C00 | PB port configuration register low | 0x44444444 |
| R32_GPIOB_CFGHR | 0x40010C04 | PB port configuration register high | 0x44444444 |
| R32_GPIOC_CFGLR | 0x40011000 | PC port configuration register low | 0x00444444 |
| R32_GPIOA_INDR | 0x40010808 | PA port input data register | 0x0000XXXX |
| R32_GPIOB_INDR | 0x40010C08 | PB port input data register | 0x0000XXXX |
| R32_GPIOC_INDR | 0x40011008 | PC port input data register | 0x000000XX |
| R32_GPIOA_OUTDR | 0x4001080C | PA port output data register | 0x00000000 |
| R32_GPIOB_OUTDR | 0x40010C0C | PB port output data register | 0x00000000 |
| R32_GPIOC_OUTDR | 0x4001100C | PC port output data register | 0x00000000 |
| R32_GPIOA_BSHR | 0x40010810 | PA port set/reset register | 0x00000000 |
| R32_GPIOB_BSHR | 0x40010C10 | PB port set/reset register | 0x00000000 |
| R32_GPIOC_BSHR | 0x40011010 | PC port set/reset register | 0x00000000 |
| R32_GPIOA_BCR | 0x40010814 | PA port reset register | 0x00000000 |
| R32_GPIOB_BCR | 0x40010C14 | PB port reset register | 0x00000000 |
| R32_GPIOC_BCR | 0x40011014 | PC port reset register | 0x00000000 |
| R32_GPIOA_LCKR | 0x40010818 | PA port lock configuration register | 0x00000000 |

| | | | |
|---|---|---|---|
| R32_GPIOB_LCKR | 0x40010C18 | PB port lock configuration register | 0x00000000 |
| R32_GPIOC_LCKR | 0x40011018 | PC port lock configuration register | 0x00000000 |

#### 6.3.1.1 Port Configuration Register Low (GPIOx_CFGLR) (x=A/B/C)

Offset address: 0x00

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| CNF7[1:0] | MODE7[1:0] | CNF6[1:0] | MODE6[1:0] | CNF5[1:0] | MODE5[1:0] | CNF4[1:0] | MODE4[1:0] |
| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
| CNF3[1:0] | MODE3[1:0] | CNF2[1:0] | MODE2[1:0] | CNF1[1:0] | MODE1[1:0] | CNF0[1:0] | MODE0[1:0] |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:30]<br>[27:26]<br>[23:22]<br>[19:18]<br>[15:14]<br>[11:10]<br>[7:6]<br>[3:2] | CNFy[1:0] | RW | GPIOA(y=0-7), GPIOB(y=0-7), GPIOC(y=0-5), the configuration bits for port x, by which the corresponding port is configured.<br>When in input mode (MODE=00b).<br>00: Analog input mode.<br>01: Floating input mode.<br>10: With pull-up mode, only some IO supports controllable pull-down.<br>11: Reserved.<br>In output mode (MODE>00b).<br>00: General-purpose push-pull output mode.<br>01: General-purpose open-drain output mode;<br>10: Alternate function push-pull output mode.<br>11: Alternate function open-drain output mode. | 01b |
| [29:28]<br>[25:24]<br>[21:20]<br>[17:16]<br>[13:12]<br>[9:8]<br>[5:4]<br>[1:0] | MODEy[1:0] | RW | GPIOA(y=0-7), GPIOB(y=0-7), GPIOC(y=0-5), port x mode selection, configure the corresponding port by these bits.<br>00: Input mode.<br>01/10/11: Output mode, max. speed 30MHz. | 00b |

*Note: PB7 only supports high-voltage open-drain output, not input.*

#### 6.3.1.2 Port Configuration Register High (GPIOx_CFGHR) (x=A/B)

Offset address: 0x04

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| CNF15[1:0] | MODE15[1:0] | CNF14[1:0] | MODE14[1:0] | CNF13[1:0] | MODE13[1:0] | CNF12[1:0] | MODE12[1:0] |
| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
| CNF11[1:0] | MODE11[1:0] | CNF10[1:0] | MODE10[1:0] | CNF9[1:0] | MODE9[1:0] | CNF8[1:0] | MODE8[1:0] |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:30]<br>[27:26]<br>[23:22] | CNFy[1:0] | RW | GPIOA(y=8-15), GPIOB(y=8-15), the configuration bits for port x, by which the corresponding port is configured. | 01b |

| | | | | |
|---|---|---|---|---|
| [19:18]<br>[15:14]<br>[11:10]<br>[7:6]<br>[3:2] | | | When in input mode (MODE=00b).<br>00: Analog input mode.<br>01: Floating input mode.<br>10: With pull-up and pull-down mode.<br>11: Reserved.<br>In output mode (MODE>00b).<br>00: General-purpose push-pull output mode.<br>01: General-purpose open-drain output mode;<br>10: Alternate function push-pull output mode.<br>11: Alternate function open-drain output mode. | |
| [29:28]<br>[25:24]<br>[21:20]<br>[17:16]<br>[13:12]<br>[9:8]<br>[5:4]<br>[1:0] | MODEy[1:0] | RW | GPIOA(y=8-15), GPIOB(y=8-15), port x mode selection, configure the corresponding port by these bits.<br>00: Input mode.<br>01/10/11: Output mode, max. speed 30MHz. | 00b |

*Note: PB9, PB11, PB13 and PB15 only support push-pull output, and output is low by default, and input is not supported.*

### 6.3.1.3 Port Input Data Register (GPIOx_INDR) (x=A/B/C)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IDR15 | IDR14 | IDR13 | IDR12 | IDR11 | IDR10 | IDR9 | IDR8 | IDR7 | IDR6 | IDR5 | IDR4 | IDR3 | IDR2 | IDR1 | IDR0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0 |
| [15:0] | IDRy | RO | (y=0-15), the port input data. These bits are read-only and can only be read as 16 bits. The read value is the high and low state of the corresponding bit. | X |

### 6.3.1.4 Port Output Register (GPIOx_OUTDR) (x=A/B/C)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ODR15 | ODR14 | ODR13 | ODR12 | ODR11 | ODR10 | ODR9 | ODR8 | ODR7 | ODR6 | ODR5 | ODR4 | ODR3 | ODR2 | ODR1 | ODR0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0 |
| [15:0] | ODRy | RW | (y=0-15), the data output by the port. These data can only be manipulated in the form of 16 bits. The IO port outputs the values of these registers. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | For input modes with pull-up and pull-down:<br>0: Pull-down input;<br>1: Pull-down input, only some IO supports controllable pull-down. | |

### 6.3.1.5 Port Reset/Set Register (GPIOx_BSHR) (x=A/B/C)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BS15 | BS14 | BS13 | BS12 | BS11 | BS10 | BS9 | BS8 | BS7 | BS6 | BS5 | BS4 | BS3 | BS2 | BS1 | BS0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | BRy | WO | (y=0-15), the corresponding OUTDR bit will be cleared for these location bits, writing 0 has no effect. These bits can only be accessed in 16 bits. If both the BR and BS bits are set, the BS bits work. | 0 |
| [15:0] | BSy | WO | (y=0-15), for these position bits will make the corresponding OUTDR position bit, writing 0 has no effect. These bits can only be accessed in 16 bits. If both the BR and BS bits are set, the BS bits work. | 0 |

### 8.3.1.6 Port Reset Register (GPIOx_BCR) (x=A/B/C)

Offset address: 0x14

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0 |
| [15:0] | BRy | WO | (y=0-15), the corresponding OUTDR bit will be cleared for these location bits, writing 0 has no effect. These bits can only be accessed in 16 bits. | 0 |

### 6.3.1.7 Port Lock Configuration Register (GPIOx_LCKR) (x=A/B/C)

Offset address: 0x18

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LCKK |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LCK15 | LCK14 | LCK13 | LCK12 | LCK11 | LCK10 | LCK9 | LCK8 | LCK7 | LCK6 | LCK5 | LCK4 | LCK3 | LCK2 | LCK1 | LCK0 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 24 | LCKK | RO | The lock key, which can be written in a specific sequence to achieve locking, but which can be read out at any time. It reads 0 to indicate that no locking is in effect, and reads 1 to indicate that locking is in effect.<br>The write sequence for the lock key is: write 1 - write 0 - write 1 - read 0 - read 1. The last step is not necessary, but can be used to confirm that the lock key is active.<br>Any error while writing the sequence will not enable the activation of the lock and the value of LCK[15:0] cannot be changed while the sequence is being written. After the lock is in effect, the port configuration can only be changed after the next reset. | 0 |
| [23:0] | LCKy | RW | (y=0-15), these bits are 1 to indicate locking the configuration of the corresponding port. These bits can only be changed before the LCKK is unlocked. The locked configuration refers to the configuration registers GPIOx_CFGLR and GPIOx_CFGHR. | 0 |

*Note: After the LOCK sequence is executed for the corresponding port bit, the configuration of the port bit will not be changed again until the next system reset.*

### 6.3.2 AFIO Register Description

Unless otherwise specified, AFIO registers must be operated as words (Operate these registers with 32 bits).

Table 6-17 List of AFIO-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_AFIO_PCFR1 | 0x40010004 | Remap Register 1 | 0x00000000 |
| R32_AFIO_EXTICR1 | 0x40010008 | External interrupt configuration register 1 | 0x00000000 |
| R32_AFIO_EXTICR2 | 0x4001000C | External interrupt configuration register 2 | 0x00000000 |
| R32_AFIO_CTLR | 0x40010018 | Control register | 0x00000045 |

#### 6.3.2.1 Remap Register 1 (AFIO_PCFR1)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | SWCFG[2:0] | | | Reserved | | | HO_DIO_EN3 | HO_DIO_EN2 | HO_DIO_EN1 | HO_DIO_EN0 | PB5PB6_RM |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADC_ETRGIN_RM | TIM3_RM[2:0] | | | TIM2_RM[1:0] | | TIM1_RM[2:0] | | | SPI1_RM[1:0] | | UART1_RM[2:0] | | | I2C1_RM[1:0] | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:27] | Reserved | RO | Reserved. | 0 |
| [26:24] | SWCFG[2:0] | RW | These bits are used to configure the I/O ports for SW function and trace function. SWD (SDI) is the debug interface to access the core. It is always used as a SWD | 0 |

| | | | | |
|---|---|---|---|---|
| | | | port after system reset.<br>0xx: Enable SWD (SDI).<br>100: Disable SWD (SDI), which functions as a GPIO.<br>Others: Invalid. | |
| [23:21] | Reserved | RO | Reserved | 0 |
| 20 | HO_DIO_EN3 | RW | PB15 $V_{DD8}$ diode output to $V_B$ enable:<br>1: Always the diode output;<br>0: Pass-through/diode output is optional. | 0 |
| 19 | HO_DIO_EN2 | RW | PB13 $V_{DD8}$ diode output to $V_B$ enable:<br>1: Always the diode output;<br>0: Pass-through/diode output is optional. | 0 |
| 18 | HO_DIO_EN1 | RW | PB11 $V_{DD8}$ diode output to $V_B$ enable:<br>1: Always the diode output;<br>0: Pass-through/diode output is optional. | 0 |
| 17 | HO_DIO_EN0 | RW | PB9 $V_{DD8}$ diode output to $V_B$ enable:<br>1: Always the diode output;<br>0: Pass-through/diode output is optional. | 0 |
| 16 | PB5PB6_RM | RW | Pins PB5&PB6 remap bits, which can be read and written by users. It controls whether the GPIO functions of PB5 and PB6 are remapped, that is, PB5&PB6 are mapped to OSC_IN&OSC_OUT.<br>1: The pin is used as a crystal oscillator pin;<br>0: The pin is used as a GPIO port. | 0 |
| 15 | ADC_ETRGIN_RM | RW | The remapping bit of the external trigger conversion of ADC.<br>1: ADC external trigger conversion is connected with PB6;<br>0: ADC external trigger conversion is connected to PA14. | 0 |
| [14:12] | TIM3_RM[2:0] | RW | The remap bit of Timer3. These bits can be read and written by the user. It controls the mapping of channels 1 to 2, 1N and 2N of timer 3 in GPIO port.<br>000: Default mapping (CH1/PC0, CH2/PC1, CH1N/PB2, CH2N/PB3);<br>001: Partial mapping (CH1/PC0, CH2/PB2, CH1N/PC1, CH2N/PB3);<br>010: Full mapping (CH1/PB2, CH2/PB3, CH1N/PB4, CH2N/PB5);<br>011: Full mapping (CH1/PA2, CH2/PB5, CH1N/PA4, CH2N/PB6);<br>100: Full mapping (CH1/PB13, CH2/PB15, CH1N/PB12, CH2N/PB14);<br>Others: Reserved. | 0 |
| [11:10] | TIM2_RM[1:0] | RW | Remap bits for timer 2. These bits can be read and written by the user. It controls the mapping of channels 1 to 4, 1N and 2N of Timer 2 and external trigger (ETR) on GPIO port.<br>00: Default mapping (CH1/ETR/PA5, CH2/PA6, CH3/CH1N/PA7, CH4/CH2N/PA4);<br>01: Partial mapping (CH1/ETR/PA3, CH2/PB0, CH3/CH1N/PB1 CH4/CH2N/PB2);<br>10: Full mapping (CH1/ETR/PB13, CH2/PB15, CH3/CH1N/PB12, CH4/CH2N/PB14);<br>11: Full mapping (CH1/ETR/PB5, CH2/PB6, CH3/CH1N/PB1, CH4/CH2N/PB2). | 0 |
| [9:7] | TIM1_RM[2:0] | RW | Remap bits for timer 1. These bits can be read and written by the user. It controls the mapping of channels | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1 to 4, 1N to 3N, external trigger (ETR) and brake input (BKIN) of Timer 1 in GPIO port.<br>000: Default mapping (ETR/PC1, CH1/PB9, CH2/PB11, CH3/PB13, CH4/PC0, BKIN/PA15, CH1N/PB8, CH2N/PB10, CH3N/PB12);<br>001: Partial mapping (ETR/PC1, CH1/PB9, CH2/PB11, CH3/PB13, CH4/PB15, BKIN/PA13, CH1N/PB8, CH2N/PB10m CH3N/PB12);<br>010: Partial mapping (ETR/PA4, CH1/PB9, CH2/PB11, CH3/PB15, CH4/PC2, BKIN/PA15, CH1N/PB8, CH2N/PB10, CH3N/PB14);<br>011: Partial mapping (ETR/PA4, CH1/PC3, CH2/PC4, CH3/PA0, CH4/PC5, BKIN/PA15, CH1N/PC2, CH2N/PC1, CH3N/PC0);<br>100: Partial mapping (ETR/PA4, CH1/PC3/PB9, CH2/PC4/PB11, CH3/PA0/PB13, CH4/PA1, BKIN/PA15, CH1N/PC2/PB8, CH2N/PC1/PB10, CH3N/PC0/PB12);<br>Others: Reserved. | |
| [6:5] | SPI1_RM[1:0] | RW | Remapping of SPI1. This bit can be read and written by the user. It controls the mapping of NSS, SCK, MISO and MOSI multiplexing functions of SPI1 on GPIO ports.<br>00: Default mapping (NSS/PA0, SCK/PA1, MISO/PC4, MOSI/PC3);<br>01: Remapping (NSS/PA2, SCK/PB6, MISO/PA4, MOSI/PB5);<br>10: Remapping (NSS/PA0, SCK/PA1, MISO/PC4, MOSI/PA11);<br>11: Remapping (NSS/PA2, SCK/PA1, MISO/PA4, MOSI/PA8). | 0 |
| [4:2] | UART1_RM[2:0] | RW | Remap bit for USART1.<br>000: Default mapping (TX/PC1, RX/PC0, CTS/PA14, RTS/PA15);<br>001: Remapping (TX/PC1, RX/PC2, CTS/PA14, RTS/PA15);<br>010: Remapping (TX/PB6, RX/PB5, CTS/PA2, RTS/PA4);<br>011: Remapping (TX/PB1, RX/PB0, CTS/PB2, RTS/PB3);<br>100: Remapping (TX/PA3, RX/PA2, CTS/PB2, RTS/PB3);<br>101: Remapping (TX/PA2, RX/PA3, CTS/PB2, RTS/PB3);<br>Others: Reserved. | 0 |
| [1:0] | I2C1_RM[2:0] | RW | Remap bit for I2C1.<br>00: Default mapping (SCL/PB3, SDA/PB2);<br>01: Remapping (SCL/PA2, SDA/PB6);<br>10: Remapping (SCL/PA15, SDA/PA14);<br>11: Remapping (SCL/PA2, SDA/PA3). | 0 |

### 6.3.2.2 External Interrupt Configuration Register 1 (AFIO_EXTICR1)

Offset address: 0x08

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| EXTI15 | EXTI14 | EXTI13 | EXTI12 | EXTI11 | EXTI10 | EXTI9 | EXTI8 |

| [1:0] | [1:0] | [1:0] | [1:0] | [1:0] | [1:0] | [1:0] | [1:0] |
|---|---|---|---|---|---|---|---|
| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
| EXTI7[1:0] | EXTI6[1:0] | EXTI5[1:0] | EXTI4[1:0] | EXTI3[1:0] | EXTI2[1:0] | EXTI1[1:0] | EXTI0[1:0] |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:30]<br>[29:28]<br>[27:26]<br>[25:24]<br>[23:22]<br>[21:20]<br>[19:18]<br>[17:16]<br>[15:14]<br>[13:12]<br>[11:10]<br>[9:8]<br>[7:6]<br>[5:4]<br>[3:2]<br>[1:0] | EXTIx[1:0] | RW | (x=0-15) external interrupt input pin configuration bit. Used to determine to which port pins the external interrupt pins are mapped.<br>00: xth pin of the PA pin, x=0-15.<br>01: xth pin of the PB pin, x=0-15.<br>10: xth pin of the PC pin, x=0-5.<br>11: Reserved. | 0 |

# Chapter 7 Direct Memory Access Control (DMA)

The Direct Memory Access Controller (DMA) provides a high-speed means of data transfer between peripherals and memory or between memory and memory without CPU intervention, data can be moved quickly through the DMA to save CPU resources for other operations.

The DMA controller has 7 channels, each channel of the DMA controller is dedicated to managing requests for memory access from one or more peripherals. There is also an arbiter to co-ordinate priorities between the channels.

## 7.1 Main Features

- 7 general-purpose independently configurable channels
- Each channel is directly connected to a dedicated hardware DMA request and supports software triggering
- Support Buffer management with loop
- Request priority between multiple channels can be set by software programming (Very high, high, medium and low) and priority setting is determined by the channel number when equal (The lower the channel number the higher the priority)
- Support peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfers
- Flash, SRAM, peripheral SRAM and HB peripherals can be used as access sources and targets
- Programmable number of data transfer bytes: up to 65535

## 7.2 Function Description

### 7.2.1 DMA Channel Processing

**1) Arbitration priority**

DMA requests generated by 7 independent channels are fed to the DMA controller via a logical or structure, and only one channel request is currently responded to. An arbiter inside the module selects the peripheral/memory access to be initiated based on the priority of the channel request.

In software management, the application can configure the priority level for each channel independently by setting the PL[1:0] bits of the DMA_CFGRx register, including 4 levels: highest, high, medium and low. When the software setting levels are the same between channels, the module will be selected according to a fixed hardware priority, with the lower channel number having a higher priority than the higher one.

**2) DMA configuration**

When the DMA controller receives a request signal, it accesses the requested peripheral or memory and establishes a data transfer between the peripheral or memory and the memory. It consists of the following 3 main operation steps.

(1) Fetch data from the memory address indicated by the Peripheral Data Register or the Current Peripheral/Memory Address Register. The start address for the first transfer is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx registers.

(2) Store data to the memory address indicated by the Peripheral Data Register or the Current Peripheral/Memory Address Register, and the start address for the first transfer is the peripheral base address or memory address

specified by the DMA_PADDRx or DMA_MADDRx registers.

(3) Performs a decrement operation of the value in the DMA_CNTRx register, which indicates the number of operations currently outstanding for transfer.

Each channel includes 3 types of DMA data transfer methods.

- Peripheral to memory (MEM2MEM=0, DIR=0)
- Memory to peripheral (MEM2MEM=0, DIR=1)
- Memory to memory (MEM2MEM=1)

*Note: The memory-to-memory mode does not require a peripheral request signal. After configuring this mode (MEM2MEM=1), the channel is turned on (EN=1) to start data transfer. This mode does not support cyclic mode.*

The configuration process is as follows.

1) Set the first address of the peripheral register or the memory data address in the memory-to-memory mode (MEM2MEM=1) in the DMA_PADDRx register. This address will be the source or destination address for data transfer when a DMA request occurs.
2) Set the memory data address in the DMA_MADDRx register. When a DMA request occurs, the transferred data will be read from or written to this address.
3) Set the amount of data to be transferred in the DMA_CNTRx register. This value is decremented after each data transfer.
4) Set the priority of the channel in the PL[1:0] bits of the DMA_CFGRx register.
5) Set the direction of data transfer, cyclic mode, incremental mode for peripheral and memory, data width for peripheral and memory, transfer halfway, transfer complete, and transfer error interrupt enable bits in the DMA_CFGRx register.
6) Set the ENABLE bit of the DMA_CCRx register to start channel x (x=1/2/3/4/5/6/7).

*Note: The control bits for the direction of data transfer (DIR), cyclic mode (position), and incremental mode of peripherals and memory (MINC/PINC) in the DMA_PADDRx/DMA_MADDRx/DMA_CNTRx registers and in the DMA_CFGRx register can be configured to be written only when the DMA channel is turned off.*

**3) Circular mode**

Setting CIRC position 1 of the DMA_CFGRx register enables the cyclic mode function for channel data transfers. In cyclic mode, when the number of data transfers becomes 0, the contents of the DMA_CNTRx register are automatically reloaded to its initial value, and the internal peripheral and memory address registers are reloaded to the initial address values set by the DMA_PADDRx and DMA_MADDRx registers, and DMA operation will continue until the channel is turned off or the DMA mode is turned off.

**4) DMA processing status**

- Transfer half: It corresponds to the hardware setting of HTIFx bit in DMA_INTFR register. The DMA transfer bytes half flag will be generated when the number of DMA transfers is reduced to less than half of the initial set value, and an interrupt will be generated if HTIE is set in the DMA_CCRx register. The hardware uses this flag to alert the application that it can prepare for a new round of data transfers.
- Transfer completion: corresponds to the hardware setting of the TCIFx bit in the DMA_INTFR register. When the number of DMA transfer bytes decreases to 0, the DMA transfer completion flag will be generated, and if TCIE is set in the DMA_CCRx register, an interrupt will be generated.
- Transfer error: corresponds to a hardware set of the TEIFx bit in the DMA_INTFR register. Reading and

writing a reserved address area will generate a DMA transfer error. At the same time the module hardware will automatically clear the EN bit of the DMA_CCRx register corresponding to the channel where the error occurred, and the channel is turned off. If TEIE is set in the DMA_CCRx register, an interrupt will be generated.

When the application queries the DMA channel status, it can first access the GIFx bit of the DMA_INTFR register to determine which channel is currently experiencing a DMA event, and then process the specific DAM event content for that channel.

### 7.2.2 Programmable Total Data Transfer Size/Data Bit Width/Alignment

The total size of the data to be transferred per DMA channel round is programmable up to 65535 times, and the number of pending transfer bytes is indicated in the DMA_CNTRx register. At EN=0, the set value is written, and at EN=1 when the DMA transfer channel is turned on, this register becomes a read-only attribute with a decreasing value after each transfer.

The transferred data fetch values of peripherals and memories support the address pointer auto-increment function with programmable pointer increments. The first transmitted data address they access is stored in the DMA_PADDRx and DMA_MADDRx registers.By setting the PINC bit or MINC position 1 of the DMA_CFGRx register, the peripheral address self-increment mode or memory address self-increment mode can be enabled, respectively. PSIZE[1:0] sets the peripheral address fetch data size and address self-increment size. MSIZE[1:0] sets the memory address to take the data size and address self-increasing small, including three choices: 8-bit, 16-bit, 32-bit. The specific data transfer methods are listed in the following table.

Table 7-1 DMA transfer with different data bit widths (PINC=MINC=1)

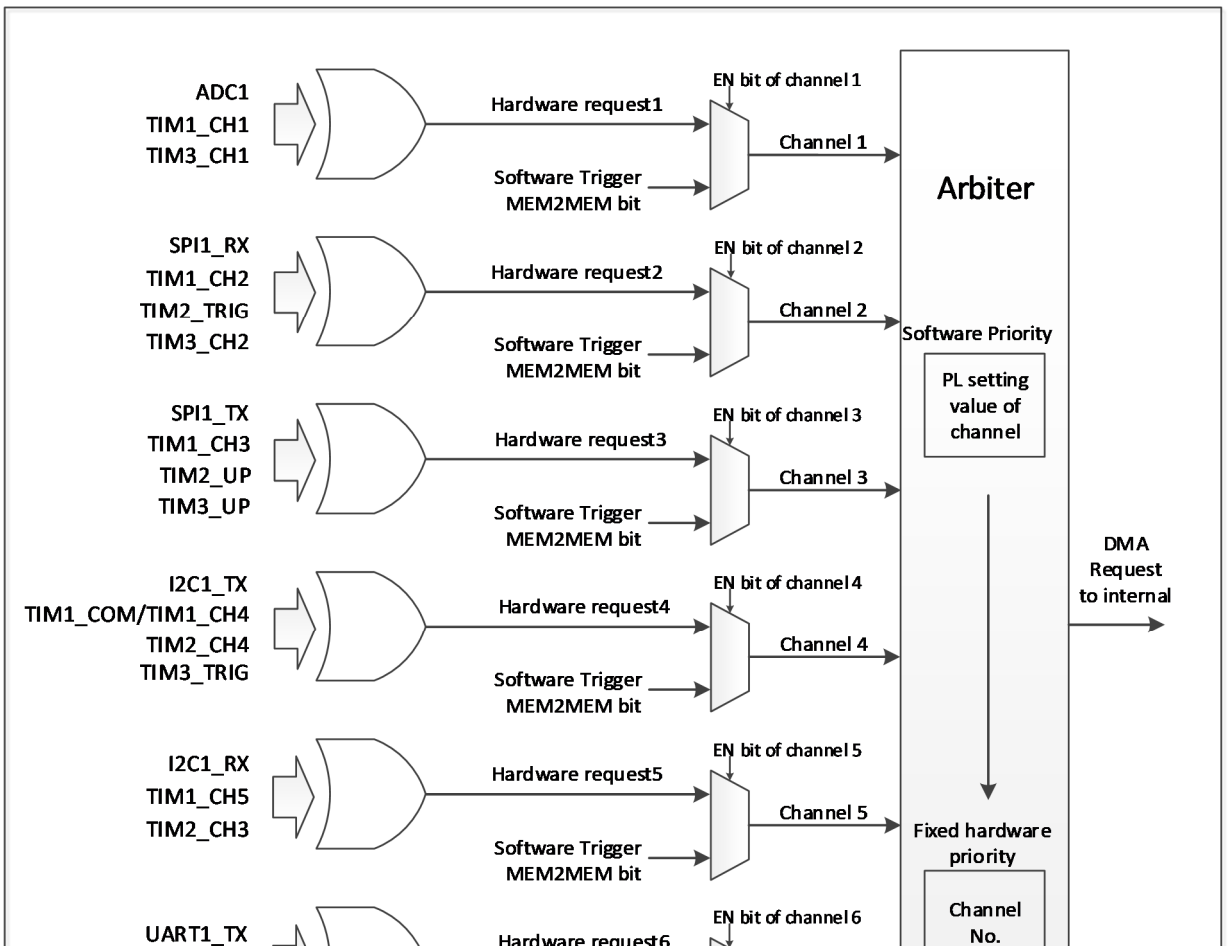
| Source bit width | Objectives bit width | Transmission number | Source: address/data | Target: address/data | Transfer operations |
|---|---|---|---|---|---|
| 8 | 8 | 4 | 0x00/B0<br>0x01/B1<br>0x02/B2<br>0x03/B3 | 0x00/B0<br>0x01/B1<br>0x02/B2<br>0x03/B3 | ● The source address increment is aligned with the data bit width set at the source and takes a value equal to the data bit width at the source<br>● The target address increment is aligned with the bit width of the target setup data and takes a value equal to the target data bit width<br>● DMA transfer of data sent to the target based on the principle: the high bit of the data size is not enough to make up 0, the high bit of the data size overflow is removed<br>● Storage data mode: small-end mode, low address stores low bytes, high address stores high bytes |
| 8 | 16 | 4 | 0x00/B0<br>0x01/B1<br>0x02/B2<br>0x03/B3 | 0x00/00B0<br>0x02/00B1<br>0x04/00B2<br>0x06/00B3 | |
| 8 | 32 | 4 | 0x00/B0<br>0x01/B1<br>0x02/B2<br>0x03/B3 | 0x00/000000B0<br>0x04/000000B1<br>0x08/000000B2<br>0x0C/000000B3 | |
| 16 | 8 | 4 | 0x00/B1B0<br>0x02/B3B2<br>0x04/B5B4<br>0x06/B7B6 | 0x00/B0<br>0x01/B1<br>0x02/B2<br>0x03/B3 | |
| 16 | 16 | 4 | 0x00/B1B0<br>0x02/B3B2<br>0x04/B5B4<br>0x06/B7B6 | 0x00/B1B0<br>0x02/B3B2<br>0x04/B5B4<br>0x06/B7B6 | |
| 16 | 32 | 4 | 0x00/B1B0<br>0x02/B3B2<br>0x04/B5B4<br>0x06/B7B6 | 0x00/0000B1B0<br>0x04/0000B3B2<br>0x08/0000B5B4<br>0x0C/0000B7B6 | |
| 32 | 8 | 4 | 0x00/B3B2B1B0<br>0x04/B7B6B5B4<br>0x08/BBBAB9B8 | 0x00/B0<br>0x01/B1<br>0x02/B2 | |

| | | | | | |
|---|---|---|---|---|---|
| | | | 0x0C/BFBEBDBC | 0x03/B3 | |
| 32 | 16 | 4 | 0x00/B3B2B1B0<br>0x04/B7B6B5B4<br>0x08/BBBAB9B8<br>0x0C/BFBEBDBC | 0x00/B1B0<br>0x02/B3B2<br>0x04/B5B4<br>0x06/B7B6 | |
| 32 | 32 | 4 | 0x00/B3B2B1B0<br>0x04/B7B6B5B4<br>0x08/BBBAB9B8<br>0x0C/BFBEBDBC | 0x00/B3B2B1B0<br>0x04/B7B6B5B4<br>0x08/BBBAB9B8<br>0x0C/BFBEBDBC | |

### 7.2.3 DMA Request Mapping

The DMA controller provides 7 channels, each of which corresponds to multiple peripheral requests. By setting the corresponding DMA control bits in the corresponding peripheral registers, the DMA function of each peripheral can be independently turned on or off.

Figure 7-1 DMA request image




| Peripheral | Channel1 | Channel 2 | Channel 3 | Channel 4 | Channel 5 | Channel 6 | Channel 7 |
|---|---|---|---|---|---|---|---|
| ADC | ADC | | | | | | |
| SPI1 | | SPI1_RX | SPI1_TX | | | | |
| UART1 | | | | | | UART1_TX | UART1_RX |
| I2C1 | | | | I2C1_TX | I2C1_RX | | |
| TIM1 | TIM1_CH1 | TIM1_CH2 | TIM1_CH3 | TIM1_CH4 | TIM1_CH5 | TIM1_UP | TIM1_TRIG |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | TIM1_COM | | | |
| TIM2 | | TIM2_TRIG | TIM2_UP | TIM2_CH4 | TIM2_CH3 | TIM2_CH2 | TIM2_CH1 |
| TIM3 | TIM3_CH1 | TIM3_CH2 | TIM3_UP | TIM3_TRIG | | | |

## 7.3 Register Description

Table 7-2 DMA-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_DMA_INTFR | 0x40020000 | DMA interrupt status register | 0x00000000 |
| R32_DMA_INTFCR | 0x40020004 | DMA interrupt flag clear register | 0x00000000 |
| R32_DMA_CFGR1 | 0x40020008 | DMA channel 1 configuration register | 0x00000000 |
| R32_DMA_CNTR1 | 0x4002000C | DMA channel 1 transfer data number register | 0x00000000 |
| R32_DMA_PADDR1 | 0x40020010 | DMA channel 1 peripheral address register | 0x00000000 |
| R32_DMA_MADDR1 | 0x40020014 | DMA channel 1 memory address register | 0x00000000 |
| R32_DMA_CFGR2 | 0x4002001C | DMA channel 2 configuration register | 0x00000000 |
| R32_DMA_CNTR2 | 0x40020020 | DMA channel 2 transfer data number register | 0x00000000 |
| R32_DMA_PADDR2 | 0x40020024 | DMA channel 2 peripheral address register | 0x00000000 |
| R32_DMA_MADDR2 | 0x40020028 | DMA channel 2 memory address register | 0x00000000 |
| R32_DMA_CFGR3 | 0x40020030 | DMA channel 3 configuration register | 0x00000000 |
| R32_DMA_CNTR3 | 0x40020034 | DMA channel 3 transfer data number register | 0x00000000 |
| R32_DMA_PADDR3 | 0x40020038 | DMA channel 3 peripheral address register | 0x00000000 |
| R32_DMA_MADDR3 | 0x4002003C | DMA channel 3 memory address register | 0x00000000 |
| R32_DMA_CFGR4 | 0x40020044 | DMA channel 4 configuration register | 0x00000000 |
| R32_DMA_CNTR4 | 0x40020048 | DMA channel 4 transfer data number register | 0x00000000 |
| R32_DMA_PADDR4 | 0x4002004C | DMA channel 4 peripheral address register | 0x00000000 |
| R32_DMA_MADDR4 | 0x40020050 | DMA channel 4 memory address register | 0x00000000 |
| R32_DMA_CFGR5 | 0x40020058 | DMA channel 5 configuration register | 0x00000000 |
| R32_DMA_CNTR5 | 0x4002005C | DMA channel 5 transfer data number register | 0x00000000 |
| R32_DMA_PADDR5 | 0x40020060 | DMA channel 5 peripheral address register | 0x00000000 |
| R32_DMA_MADDR5 | 0x40020064 | DMA channel 5 memory address register | 0x00000000 |
| R32_DMA_CFGR6 | 0x4002006C | DMA channel 6 configuration register | 0x00000000 |
| R32_DMA_CNTR6 | 0x40020070 | DMA channel 6 transfer data number register | 0x00000000 |
| R32_DMA_PADDR6 | 0x40020074 | DMA channel 6 peripheral address register | 0x00000000 |
| R32_DMA_MADDR6 | 0x40020078 | DMA channel 6 memory address register | 0x00000000 |
| R32_DMA_CFGR7 | 0x40020080 | DMA channel 7 configuration register | 0x00000000 |
| R32_DMA_CNTR7 | 0x40020084 | DMA channel 7 transfer data number register | 0x00000000 |
| R32_DMA_PADDR7 | 0x40020088 | DMA channel 7 peripheral address register | 0x00000000 |
| R32_DMA_MADDR7 | 0x4002008C | DMA channel 7 memory address register | 0x00000000 |

### 7.3.1 DMA Interrupt Status Register (DMA_INTFR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | TEIF7 | HTIF7 | TCIF7 | GIF7 | TEIF6 | HTIF6 | TCIF6 | GIF6 | TEIF5 | HTIF5 | TCIF5 | GIF5 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TEIF4 | HTIF4 | TCIF4 | GIF4 | TEIF3 | HTIF3 | TCIF3 | GIF3 | TEIF2 | HTIF2 | TCIF2 | GIF2 | TEIF1 | HTIF1 | TCIF1 | GIF1 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:28] | Reserved | RO | Reserved | 0 |
| 27/23/19/15/11/7/3 | TEIFx | RO | Transmission error flag for channel x (x=1/2/3/4/5/6/7).<br>1: A transmission error occurred on channel x.<br>0: No transmission error on channel x.<br>Hardware set, software write CTEIFx bit to clear this flag. | 0 |
| 26/22/18/14/10/6/2 | HTIFx | RO | Transmission halfway flag for channel x (x=1/2/3/4/5/6/7).<br>1: A transmission over half event is generated on channel x.<br>0: No transmission over half on channel x.<br>Hardware set, software write CHTIFx bit to clear this flag. | 0 |
| 25/21/17/13/9/5/1 | TCIFx | RO | Transmission completion flag for channel x (x=1/2/3/4/5/6/7).<br>1: A transmission completion event is generated on channel x.<br>0: No transmission completion event on channel x.<br>Hardware set, software write CTCIFx bit to clear this flag. | 0 |
| 24/20/16/12/8/4/0 | GIFx | RO | Global interrupt flag for channel x (x=1/2/3/4/5/6/7).<br>1: TEIFx or HTIFx or TCIFx is generated on channel x.<br>0: No TEIFx or HTIFx or TCIFx occurred on channel x.<br>Hardware set, software write CGIFx bit to clear this flag. | 0 |

### 7.3.2 DMA Interrupt Flag Clear Register (DMA_INTFCR)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | CTEIF7 | CHTIF7 | CTCIF7 | CGIF7 | CTEIF6 | CHTIF6 | CTCIF6 | CGIF6 | CTEIF5 | CHTIF5 | CTCIF5 | CGIF5 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTEIF4 | CHTIF4 | CTCIF4 | CGIF4 | CTEIF3 | CHTIF3 | CTCIF3 | CGIF3 | CTEIF2 | CHTIF2 | CTCIF2 | CGIF2 | CTEIF1 | CHTIF1 | CTCIF1 | CGIF1 |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:28] | Reserved | RO | Reserved | 0 |
| 27/23/19/15/11/7/3 | CTEIFx | WO | Clear the transmission error flag for channel x (x=1/2/3/4/5/6/7).<br>1: Clear the TEIFx flag in the DMA_INTFR register.<br>0: No effect. | 0 |
| 26/22/18/14/10/6/2 | CHTIFx | WO | Clear the transmission halfway flag for channel x (x=1/2/3/4/5/6/7).<br>1: Clear the HTIFx flag in the DMA_INTFR register.<br>0: No effect. | 0 |
| 25/21/17/13/9/5/1 | CTCIFx | WO | Clear the transmission completion flag for channel x (x=1/2/3/4/5/6/7).<br>1: Clear the TCIFx flag in the DMA_INTFR register.<br>0: No effect. | 0 |
| 24/20/16/12/8/4/0 | CGIFx | WO | Clear the global interrupt flag for channel x (x=1/2/3/4/5/6/7).<br>1: Clear the TEIFx/HTIFx/TCIFx/GIFx flags in the DMA_INTFR register.<br>0: No effect. | 0 |

### 7.3.3 DMA Channel x Configuration Register (DMA_CFGRx)(x=1/2/3/4/5/6/7)

Offset address: 0x08 + (x-1)*20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | MEM2MEM | PL[1:0] | | MSIZE[1:0] | | PSIZE[1:0] | | MINC | PINC | CIRC | DIR | TEIE | HTIE | TCIE | EN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved. | 0 |
| 14 | MEM2MEM | RW | Memory-to-memory mode enable.<br>1: Enable memory-to-memory data transfer mode.<br>0: Disable memory-to-memory data transfer mode. | 0 |
| [13:12] | PL[1:0] | RW | Channel priority setting.<br>00: Low; 01: Medium.<br>10: High; 11: Very high. | 0 |
| [11:10] | MSIZE[1:0] | RW | Memory address data width setting.<br>00: 8-bit; 01: 16-bit.<br>10: 32-bit; 11: Reserved. | 0 |
| [9:8] | PSIZE[1:0] | RW | Peripheral address data width setting.<br>00: 8-bit; 01: 16-bit.<br>10: 32-bit; 11: Reserved. | 0 |
| 7 | MINC | RW | Memory address incremental increment mode enable.<br>1: Enable incremental memory address increment operation.<br>0: Memory address remains unchanged operation. | 0 |
| 6 | PINC | RW | Peripheral address incremental increment mode enable.<br>1: Enable incremental increment operation of the peripheral address.<br>0: Peripheral address remains unchanged operation. | 0 |
| 5 | CIRC | RW | DMA channel cyclic mode enable.<br>1: Enables cyclic operation.<br>0: Perform a single operation. | 0 |
| 4 | DIR | RW | Data transfer direction.<br>1: Read from memory.<br>0: Read from peripheral. | 0 |
| 3 | TEIE | RW | Transmission error interrupt enable control.<br>1: Enable transmission error interrupt.<br>0: Disable transmission error interrupt. | 0 |
| 2 | HTIE | RW | Transmission over half interrupt enable control.<br>1: Enable the transmission over half interrupt.<br>0: Disable the transmission over half interrupt. | 0 |
| 1 | TCIE | RW | Transmission completion interrupt enable control.<br>1: Enable the transmission completion interrupt.<br>0: Disable the transmission completion interrupt. | 0 |
| 0 | EN | RW | Channel enable control.<br>1: Channel on; 0: Channel off.<br>When a DMA transfer error occurs, the hardware automatically clears this bit to 0 and shuts down the channel. | 0 |

### 7.3.4 DMA Channel x Number of Data Register (DMA_CNTRx) (x=1/2/3/4/5/6/7)

Offset address: 0x0C + (x-1)*20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NDT[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0 |
| [15:0] | NDT[15:0] | RW | Number of data transfers, range 0-65535.<br>Indicates the number of remaining pending transfer bytes (The register content is decremented after each DMA transfer).<br>When the channel is in cyclic mode, the contents of the register will be automatically reloaded to the previously configured value. | 0 |

*Note: This register can only be changed when EN=0; When EN=1, it is a read-only register, indicating the current number to be transmitted. When the register content is 0, no data transmission will occur regardless of whether the channel is open or not.*

### 7.3.5 DMA Channel x Peripheral Address Register (DMA_PADDRx) (x=1/2/3/4/5/6/7)

Offset address: 0x10 + (x-1)*20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PA[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | PA[31:0] | RW | Peripheral base address, which serves as the source or destination address for peripheral data transfer.<br>When PSIZE[1:0]='01' (16 bits), the module automatically ignores bit0 and the operation address is automatically 2-byte aligned; when PSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0] and the operation address is automatically 4-byte aligned. | 0 |

*Note: This register can only be changed when EN=0 and cannot be written when EN=1.*

### 7.3.6 DMA Channel x Memory Address Register (DMA_MADDRx) (x=1/2/3/4/5/6/7)

Offset address: 0x14 + (x-1)*20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MA[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | MA[31:0] | RW | The memory data address, which serves as the source or destination address for data transfers.<br>When MSIZE[1:0]='01' (16 bits), the module automatically ignores bit0, and the operation address is automatically 2-byte aligned; when MSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0], and the operation address is automatically 4-byte aligned. | 0 |

*Note: This register can only be changed when EN=0 and cannot be written when EN=1.*

# Chapter 8 Analog-to-digital Converter (ADC)

The ADC module contains a 12-bit successive approximation type analogue-to-digital converter. It supports 20 external channels. Single conversion and continuous conversion of channels, automatic scan mode between channels, intermittent mode, external trigger mode and double sampling can be accomplished. The analog watchdog function allows monitoring whether the channel voltage is within the threshold range.

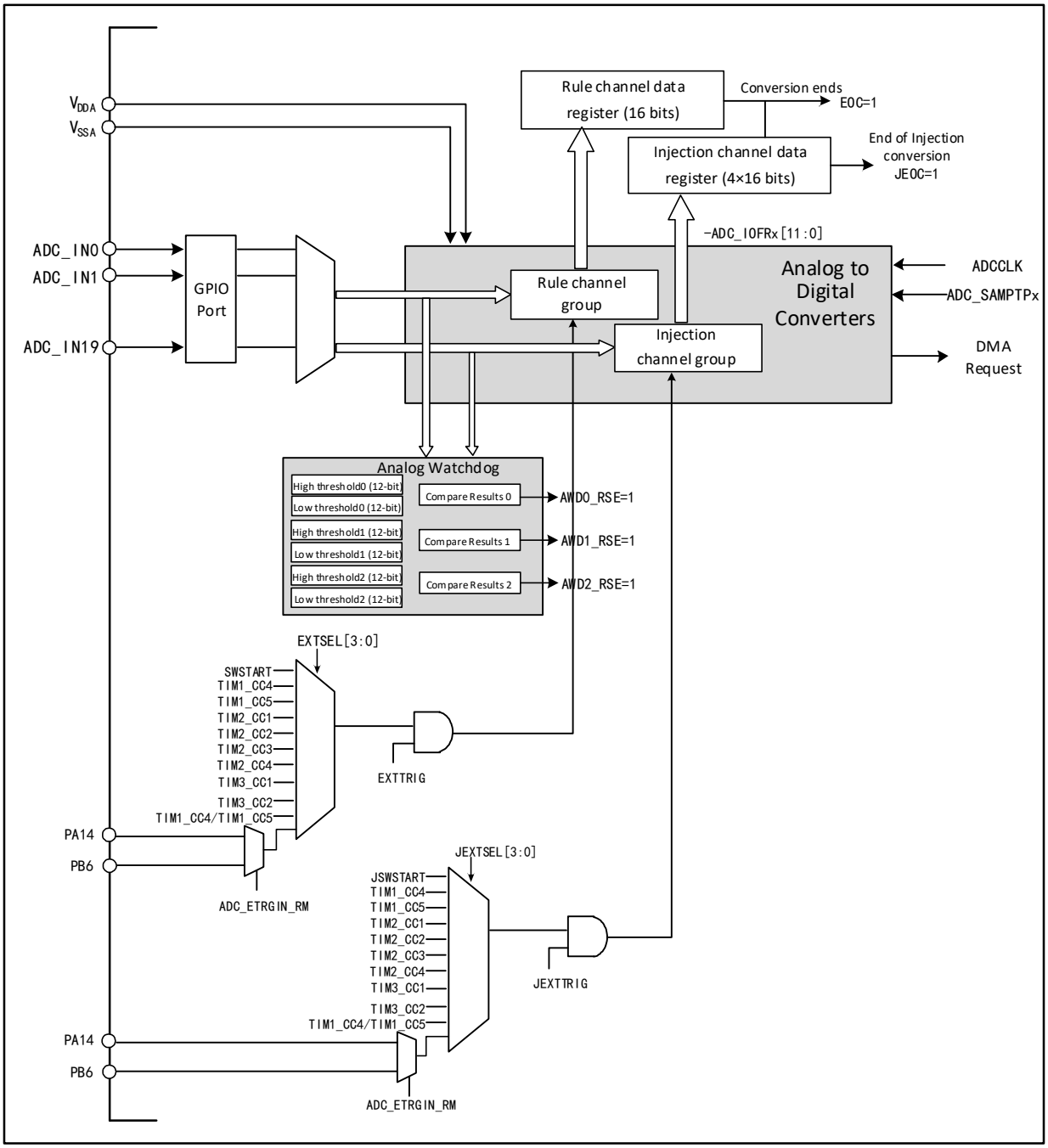
## 8.1 Main Features

- 12-bit resolution
- Support 20 external channels
- Multiple sampling conversion methods for multiple channels: Single, continuous, scan, trigger, intermittent, etc.
- Data alignment modes: left-aligned, right-aligned
- Sampling time can be programmed separately by channel
- Both rule conversion and injection conversion support external triggering
- Analog watchdog monitoring channel voltage, self-calibration function
- Analog watchdog to monitor channel voltage ADC channel input range: $0 \leq V_{IN} \leq V_{DDA}$
- Support ADC sliding average function

# 8.2 Functional Description

## 8.2.1 Module Structure

Figure 8-1 ADC module block diagram




## 8.2.2 ADC Configuration

### 1) Module power-up

An ADON bit of 1 in the ADC_CTLR2 register indicates that the ADC module is powered up. When the ADC module enters the power-up state (ADON=1) from the power-down mode (ADON=0), a delay period $t_{STAB}$ is required for the module stabilization time. After that, the ADON bit is written to 1 again and is used as the start signal for software to start the ADC conversion. By clearing the ADON bit to 0, the current conversion can be

terminated and the ADC module placed in power-down mode, a state in which the ADC consumes almost no power.

**2) Sampling clock**

The register operation of the module is based on HB bus clock, and the clock reference ADCCLK of its conversion unit is divided by ADCPRE field of RCC_CFGR0 register, and the maximum frequency cannot exceed 18MHz.

**3) Channel configuration**

The ADC module provides 20 channel sampling sources. They can be configured into 2 types of conversion groups: regular groups and injection groups. to achieve a group conversion consisting of a series of conversions in any order on any number of channels.

Conversion group:

- Rule group: consists of up to 16 conversions. The rule channels and their conversion order are set in the ADC_RSQRx register. The total number of conversions in the rule group should be written to L[3:0] in the ADC_RSQR1 register.
- Injection group: consists of up to 4 conversions. The injection channels and the order of their conversions are set in the ADC_ISQR register. The total number of conversions in the injection group should be written in JL[1:0] of the ADC_ISQR register.

*Note: If the ADC_RSQRx or ADC_ISQR registers are changed during conversion, the current conversion is terminated and a new start signal is sent to the ADC to convert the newly selected group.*

When ADC_IN9/ADC_IN10 are used as external channel inputs, it is necessary to set QDET1_EN/QDET2_EN of ISP_CTLR registers to 1.

ADC_IN9/ADC_IN10/ADC_IN18/ADC_IN19 can be multiplexed as internal channels to connect to the outputs of the OPAs, see section 17.2 for register configuration.

**4) Calibration**

The ADC has a built-in self-calibration mode. After calibration, the accuracy error caused by the change of internal capacitor bank can be greatly reduced. During calibration, an error correction code is calculated on each capacitor to eliminate the error generated on each capacitor in the subsequent conversion.

Initialize the calibration register by writing RSTCAL bit 1 in ADC_CTLR2 register, and waiting for RSTCAL hardware to clear 0 indicates that initialization is complete. Set the CAL bit to start the calibration function. Once the calibration is finished, the hardware will automatically clear the CAL bit and store the calibration code in ADC_RDATAR. After that, the normal conversion function can be started. It is recommended to perform ADC calibration once when the ADC module is powered on.

*Note: Before starting calibration, it is necessary to ensure that the ADC module is in the power-on state (ADON=1) for at least two ADC clock cycles.*

**5) Programmable sampling time**

The ADC uses several ADCCLK cycles to sample the input voltage. The number of sampling cycles for a channel can be changed using the SMPx[1:0] bits in the ADC_SAMPTR1 and ADC_SAMPTR2 registers. Each channel can be sampled separately using a different time.

The total conversion time is calculated as follows.

$$T_{CONV} = \text{sampling time} + 12.5T_{ADCCLK}$$

The ADC's rule channel conversion supports the DMA function. The value of the rule channel conversion is stored in a data-only register, ADC_RDATAR. To prevent the data in ADC_RDATAR register from being fetched in time when multiple rule channels are converted in succession, the DMA function of ADC can be enabled. The hardware will generate a DMA request at the end of the conversion of a rule channel (EOC set) and transfer the converted data from the ADC_RDATAR register to the user-specified destination address.

After the channel configuration of the DMA controller module is completed, write DMA position 1 of the ADC_CTLR2 register to enable the DMA function of the ADC.

*Note: Injection group conversion does not support DMA function.*

**6) Data alignment**

The ALIGN bit in the ADC_CTLR2 register selects the alignment of the ADC converted data storage. 12-bit data supports left-aligned and right-aligned modes.

The data register ADC_RDATAR of the rule group channel holds the actual converted 12-bit digital value; while the data register ADC_IDATARx of the injection group channel is the actual converted data minus the value written after the offset defined in the ADC_IOFRx register, there will be positive and negative cases, so there are sign bits (SIGNB).

Figure 8-2 Data left alignment

Rule group data register

| D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Inject group data register

| SIGNB | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Figure 8-3 Data right alignment

Rule group data register

| 0 | 0 | 0 | 0 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Inject group data register

| SIGNB | SIGNB | SIGNB | SIGNB | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

### 8.2.3 External Trigger Source

The ADC conversion start event can be triggered by an external event. If the EXTTRIG or JEXTTRIG bits of the ADC_CTLR2 register are set, the conversion of a rule group or injection group channel can be triggered by an external event, respectively. In this case, the configuration of EXTSEL[3:0] and JEXTSEL[3:0] bits determines the external event source for the rule group and injection group.

*Note: When the external trigger signal is selected as ADC rule or injection conversion, only its rising edge can start the conversion.*

Table 8-1 External trigger sources for rule group channels

| EXTSEL[3:0] | Trigger source | Type |
|---|---|---|
| 0000 | SWSTART bit 1 software trigger | Software control bit |
| 0001 | CC4 event of timer 1 | Internal signal from on-chip |

| | | |
|---|---|---|
| 0010 | CC5 event of timer 1 | timer |
| 0011 | CC1 event of timer 2 | |
| 0100 | CC2 event of timer 2 | |
| 0101 | CC3 event of timer 2 | |
| 0110 | CC4 event of timer 2 | |
| 0111 | CC1 event of timer 3 | |
| 1000 | CC2 event of timer 3 | |
| 1001 | PA14/PB6 event | From external pin |
| 1010 | CC4 event /CC5 event of timer 1 | Internal signal from on-chip timer |

Table 8-2 External trigger sources for injection group channels

| JEXTSEL[3:0] | Trigger source | Type |
|---|---|---|
| 0000 | JSWSTART bit 1 software trigger | Software control bit |
| 0001 | CC4 event of timer 1 | Internal signal from on-chip timer |
| 0010 | CC5 event of timer 1 | |
| 0011 | CC1 event of timer 2 | |
| 0100 | CC2 event of timer 2 | |
| 0101 | CC3 event of timer 2 | |
| 0110 | CC4 event of timer 2 | |
| 0111 | CC1 event of timer 3 | |
| 1000 | CC2 event of timer 3 | |
| 1001 | CC4 event /CC5 event of timer 1 | |
| 1010 | PA14/PB6 event | From external pin |

### 8.2.4 Conversion Mode

Table 8-3 Conversion mode combinations

| ADC_CTLR1 and ADC_CTLR2 register control bits | | | | | ADC conversion mode |
|---|---|---|---|---|---|
| CONT | SCAN | DISCEN/JDISCEN | JAUTO | Start event | |
| 0 | 0 | 0 | 0 | ADON position 1 | Single single-channel mode: A rule channel performs a single conversion. |
| | | | | External trigger method | Single single-channel mode: A single conversion is performed on one of the rule channels or injection channels. |
| | 1 | 0 | 0 | ADON position 1 or external trigger method | Single scan mode: performs a single conversion of all selected rule group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence.<br>Trigger injection method: When the rule group channel conversion process can be inserted into the injection group channel all conversion, and then continue the rule group channel conversion afterwards; but the rule group channel conversion will not be inserted when converting the injection group channel. |
| | | | 1 | ADON position 1 or external trigger method | Single scan mode: performs a single conversion of all selected rule group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence.<br>Automatic injection method: After the rule group channel is converted, the injection group channel is automatically converted. |

| | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 (DISCEN and JDISCEN cannot be 1 at the same time) | 0 | External trigger method | Single intermittent mode: Each time an event is started, a short sequence (DISCNUM[2:0] defined number) of channel number transitions is executed and cannot be restarted until all selected channel transitions are completed. *Note: The DISCEN and JDISCEN control bits are selected for the rule group and injection group respectively, and the intermittent mode cannot be configured for the rule group and injection group at the same time.* |
| | | | 1 | - | Disable this mode. |
| | 1 | 1 | X | - | No such mode. |
| 1 | 0 | 0 | 0 | ADON position 1 or external trigger method | Continuous single channel/scan mode: repeat a new round of transitions at the end of each round until CONT clears 0 to terminate. |
| | 1 | 0 | 0 | | |
| | | | 1 | | |

*Note: The external trigger events for rule groups and injection groups are different, and the 'ACON' bit can only initiate rule group channel conversion, so the initiation events for rule group and injection group channel conversion are independent.*

**1) Single single-channel conversion mode**

In this mode, only one conversion is executed for the current 1 channel. This mode performs conversion for the channel that is sorted 1st in the rule group or injection group, where it is initiated by setting ADON position 1 of the ADC_CTLR2 register (For rule channels only) or can be initiated by external trigger (For rule channels or injection channels). Once the conversion of the selected channel is completed it will.

If the conversion is for a rule group channel, the conversion data is stored in the 16-bit ADC_RDATAR register, the EOC flag is set, and an ADC interrupt is triggered if the EOCIE bit is set.

If the conversion is for an injection group channel, the conversion data is stored in the 16-bit ADC_IDATAR1 register, the EOC and JEOC flags are set, and an ADC interrupt is triggered if the JEOCIE or EOCIE bit is set.

**2) Single scan mode conversion**

The ADC scan mode is entered by setting the SCAN bit of the ADC_CTLR1 register to 1. This mode is used to scan a group of analog channels and perform a single conversion for all channels selected by ADC_RSQRx register (For regular channels) or ADC_ISQR (For injection channels) one by one, and the next channel in the same group is converted automatically when the current channel conversion is finished.

In the scan mode, there is a subdivision into triggered injection mode and automatic injection mode depending on the status of the IAUTO bit.

- Trigger injection

JAUTO bit is 0. When the trigger event of injection group channel conversion occurs during the scanning of rule group channels, the current conversion is reset and the sequence of injection channels is performed in a single scan, and the last interrupted rule group channel conversion is resumed after all selected injection group channel scanning conversions are completed.

If a rule channel start event occurs while the injection group channel sequence is currently being scanned, the injection group conversion is not interrupted, but the rule sequence conversion is executed again after the injection

sequence conversion is completed.

*Note: When using triggered injection conversions, you must ensure that the interval between triggered events is longer than the injection sequence. For example, if the overall time to complete the conversion of the injection sequence takes 28 ADCCLK, then the minimum value of the event interval to trigger the injection channel is 29 ADCCLK.*

- Auto-injection

The JAUTO bit is set to 1, and conversion of the selected channel of the injection group is performed automatically after scanning all the channels selected by the rule group for conversion. This approach can be used to convert up to 20 conversion sequences in the ADC_RSQRx and ADC_ISQR registers.

In this mode, external triggering of the injection channel must be disabled (JEXTTRIG=0).

*Note: For ADC clock prescaler factor (CLK_DIV[3:0]) of 4 to 8, 1 ADCCLK interval is automatically inserted when switching from rule conversion to injection sequence or from injection conversion to rule sequence; when ADC clock prescaler factor is 2, there is a delay of 2 ADCCLK intervals.*

**3) Single intermittent mode conversion**

The intermittent mode of the rule group or injection group is entered by setting the DISCEN or JDISCEN bit of the ADC_CTLR1 register to 1. This mode differs from scanning a complete set of channels in scan mode, but divides a set of channels into multiple short sequences, and each external trigger event will perform a short sequence of scan transitions.

The length of the short sequence n (n<=8) is defined in DISCNUM[2:0] of ADC_CTLR1 register, when DISCEN is 1, it is the interrupted mode of the rule group, and the total length to be converted is defined in RLEN[3:0] of ADC_RSQR1 register; when JDISCEN is 1, it is the interrupted mode of the injection group, and the total length to be converted is defined in ILEN[1:0] of ADC_ISQR register. It is not possible to set both the rule group and the injection group to intermittent mode.

Example of rule group intermittent mode.

DISCEN=1, DISCNUM[2:0]=3, L[3:0]=8, channels to be converted = 1, 3, 2, 5, 8, 4, 10, 6

The 1st external trigger: conversion sequence is: 1, 3, 2

The 2nd external trigger: conversion sequence is: 5, 8, 4

The 3rd external trigger: conversion sequence is: 10, 6, while generating EOC events

The 4th external trigger: conversion sequence is: 1, 3, 2

Examples of intermittent patterns injected into groups.

JDISCEN=1, DISCNUM[2:0]=1, JL[1:0]=3, channel to be converted=1, 3, 2

The 1st external trigger: conversion sequence is: 1

The 2nd external trigger: the conversion sequence is: 3

The 3rd external trigger: conversion sequence is: 2, generating both EOC and JEOC events

The 4th external trigger: conversion sequence is: 1

*Note: 1. When converting a rule group or injection group in intermittent mode, the conversion sequence does not automatically start from the beginning when it ends. When all subgroups have been converted, the next trigger event starts the conversion of the first subgroup.*

*2. You cannot use auto-injection (IAUTO=1) and intermittent mode at the same time.*

*3. You cannot set intermittent mode for both rule groups and injection groups, and intermittent mode can only*

*be used for a group of conversions.*

**4) Continuous conversion**

In the continuous conversion mode, another conversion is started as soon as the previous ADC conversion is finished, and the conversion does not stop on the last channel of the selection group, but continues again from the first channel of the selection group. The startup events for this mode include an external trigger event and ADON position 1, which is required to set the CONT position 1 after startup.

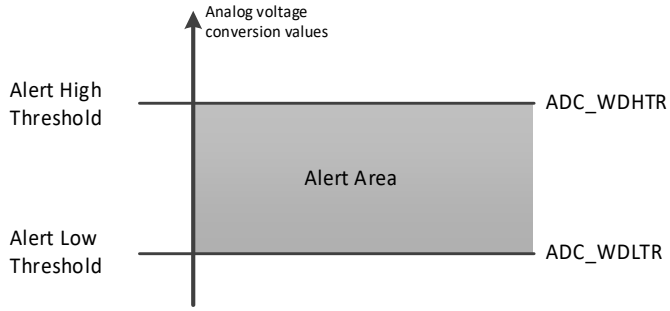
If a rule channel is converted, the conversion data is stored in the ADC_RDATAR register, and the end-of-conversion flag EOC is set, generating an interrupt if EOCIE is set.

If an injection channel is converted, the conversion data is stored in the ADC_IDATARx register, the end-of-injection-conversion flag JEOC is set, and an interrupt is generated if JEOCIE is set.

### 8.2.5 Analog Watchdog

The AWD analog watchdog status bit is set if the analog voltage being converted by the ADC is below the low threshold or above the high threshold. The threshold settings are located in the lowest 12 valid bits of the ADC_WDHTR and ADC_WDLTR registers. The AWDIE bit of the ADC_CTLR1 register is set to allow the corresponding interrupt to be generated.

Figure 8-4 Analog watchdog threshold area




Configure the AWDSGL, AWDEN, JAWDEN and AWDCH[4:0] bits of the ADC_CTLR1 register to select the channel for analog watchdog alerting, as related in the following table.

Table 8-4 Analog Watchdog channel selection

| Analog Watchdog alert channel | ADC_CTLR1 register control bit | | | |
|---|---|---|---|---|
| | AWDSGL | AWDEN | JAWDEN | AWDCH[4:0] |
| No vigilance | Ignore | 0 | 0 | Ignore |
| All injection channels | 0 | 0 | 1 | Ignore |
| All rule channels | 0 | 1 | 0 | Ignore |
| All injection and rule channels | 0 | 1 | 1 | Ignore |
| Single injection channel | 1 | 0 | 1 | Determine the channel number |
| Single rule channel | 1 | 1 | 0 | Determine the channel number |
| Single injection and rule channel | 1 | 1 | 1 | Determine the channel number |

### 8.2.6 Sliding Average Function

$$Cur_avg_adc = (Avg_adc*(K-1) + Cur_adc)/K$$

Cur_avg_adc: Adopt the current ADC slider average;

Avg_adc: Adopt the last ADC slider average;

Cur_adc: Capture the current ADC conversion value.

K: Sliding average window value.

*Note: The first Cur_avg_adc capture value is the value of Cur_adc.*

After enabling the AVG_EN enable bit of ADC_CFG register to turn on the sliding average function, the hardware calculates the sliding average according to the above formula, and the AVG_K bit sets the K value in the formula; the calculated sliding average can be read through the ADC_AVG_DR register; the high and low thresholds for the average can be set through the ADC_AVG_TR register, and the DF/UF bits of ADC_CFG register can judge the conversion value underflow (conversion value is less than the average minus the low threshold configuration value) / overflow (conversion value is greater than the average plus the high threshold configuration value); ADC_CFG register can also be used as a reference for the conversion value of the sliding average. Read the DF/UF bits in ADC_CFG register to judge the conversion value underflow (conversion value is less than the average value minus the low threshold configuration value) / overflow (conversion value is greater than the average value plus the high threshold configuration value); DFIF/UFIF bits in ADC_CFG register are the underflow / overflow interrupt flags, and corresponding interrupt can be triggered by turning on the corresponding interrupt enable bit.

*Note: The underflow/overflow constraints for determining the conversion value are: the high threshold of the sliding average + set average is less than or equal to 4095, and the low threshold of the sliding average - set average is greater than or equal to zero.*

### 8.2.7 Capture ADC Edge Function

Configure the CMP_XOR_MODE bit to select the comparison result heterodyne mode, enable the ADC_CAP bit of the TIM1_AUX1 register, the comparison result of the channel conversion/channel polling will be passed to the TIM1 channel 1 (TIM1 channel 1 is in dual-edge capture mode), and the counter will be cleared after capture.

## 8.3 Register Description

Table 8-5 ADC-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_ADC_STATR | 0x40012400 | ADC status register | 0x00000000 |
| R32_ADC_CTLR1 | 0x40012404 | ADC control register 1 | 0x00000000 |
| R32_ADC_CTLR2 | 0x40012408 | ADC control register 2 | 0x00000000 |
| R32_ADC_SAMPTR1 | 0x4001240C | ADC sample time configuration register 1 | 0x00000000 |
| R32_ADC_SAMPTR2 | 0x40012410 | ADC sample time configuration register 2 | 0x00000000 |
| R32_ADC_IOFR1 | 0x40012414 | ADC injected channel data offset register 1 | 0x00000000 |
| R32_ADC_IOFR2 | 0x40012418 | ADC injected channel data offset register 2 | 0x00000000 |
| R32_ADC_IOFR3 | 0x4001241C | ADC injected channel data offset register 3 | 0x00000000 |
| R32_ADC_IOFR4 | 0x40012420 | ADC injected channel data offset register 4 | 0x00000000 |
| R32_ADC_WDHTR | 0x40012424 | ADC watchdog high threshold register | 0x00000FFF |
| R32_ADC_WDLTR | 0x40012428 | ADC watchdog low threshold register | 0x00000000 |
| R32_ADC_RSQR1 | 0x4001242C | ADC regular sequence register 1 | 0x00000000 |
| R32_ADC_RSQR2 | 0x40012430 | ADC regular sequence register 2 | 0x00000000 |

| | | | |
|---|---|---|---|
| R32_ADC_RSQR3 | 0x40012434 | ADC regular sequence register 3 | 0x00000000 |
| R32_ADC_ISQR | 0x40012438 | ADC injected sequence register | 0x00000000 |
| R32_ADC_IDATAR1 | 0x4001243C | ADC injected data register 1 | 0x00000000 |
| R32_ADC_IDATAR2 | 0x40012440 | ADC injected data register 2 | 0x00000000 |
| R32_ADC_IDATAR3 | 0x40012444 | ADC injected data register 3 | 0x00000000 |
| R32_ADC_IDATAR4 | 0x40012448 | ADC injected data register 4 | 0x00000000 |
| R32_ADC_RDATAR | 0x4001244C | ADC regular data register | 0x00000000 |
| R32_ADC_CTLR3 | 0x40012450 | ADC control register 3 | 0x00000000 |
| R32_ADC_WDTR1 | 0x40012454 | ADC watchdog threshold 1 register | 0x0FFF0000 |
| R32_ADC_WDTR2 | 0x40012458 | ADC watchdog threshold 2 register | 0x0FFF0000 |
| R32_ADC_AVG_TR | 0x4001245C | ADC average threshold register | 0x0FFF0000 |
| R32_ADC_DLYR | 0x40012460 | ADC delay data register | 0x00000000 |
| R32_ADC_AVG_DR | 0x40012464 | ADC average data register | 0x00000000 |
| R32_ADC_RDATAR2 | 0x40012468 | ADC rule data register 2 | 0x00000000 |
| R32_ADC_RDATAR3 | 0x4001246C | ADC rule data register 3 | 0x00000000 |
| R32_ADC_RDATAR4 | 0x40012470 | ADC rule data register 4 | 0x00000000 |

### 8.3.1 ADC Status Register (ADC_STATR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | CDNFIF | COVFIF |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MULT_CMP3 | MULT_CMP2 | MULT_CMP1 | AWD_DNF | AWD_OVF | STRT | JSTRT | JEOC | EOC | AWD |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:18] | Reserved | RO | Reserved. | 0 |
| 17 | CDNFIF | WO | Clear the switching value underflow interrupt flag of the moving average function:<br>1: Invalid;<br>0: Clear the conversion value underflow interrupt flag. | 0 |
| 16 | COVFIF | WO | Clear the switch value overflow interrupt flag of the moving average function:<br>1: Invalid;<br>0: Clear the conversion value overflow interrupt flag bit. | 0 |
| [15:10] | Reserved | RO | Reserved | 0 |
| 9 | MULT_CMP3 | RO | Multi-threshold comparison result 3:<br>1: The first conversion result is greater than the 4th conversion result;<br>0: The first conversion result is less than or equal to the 4th conversion result. | 0 |
| 8 | MULT_CMP2 | RO | Multi-threshold comparison result 2:<br>1: The first conversion result is greater than the 3rd conversion result;<br>0: The first conversion result is less than or equal to the 3rd conversion result. | 0 |
| 7 | MULT_CMP1 | RO | Multi-threshold comparison result 1:<br>1: The first conversion result is greater than the 2nd conversion result;<br>0: The first conversion result is less than or equal to the 2nd conversion result. | 0 |
| 6 | AWD_DNF | RW0 | Analog watchdog underflow flag bit:<br>1: Analog watchdog underflow occurred;<br>0: No analog watchdog underflow occurred.<br>This bit is set by the hardware (the conversion value is less than ADC_WDLTR) and is cleared by the software (write 1 is invalid). | 0 |
| 5 | AWD_OVF | RW0 | Analog watchdog overflow flag bit<br>1: Analog watchdog overflow occurred;<br>0: No analog watchdog overflow occurred.<br>This bit is set by the hardware (the conversion value exceeds ADC_WDHTR) and is cleared by the software (write 1 is invalid). | 0 |
| 4 | STRT | RW0 | Rule channel transition start state.<br>1: Rule channel conversion has started.<br>0: Rule channel conversion is not started.<br>This bit is set to 1 by hardware and cleared to 0 by software (Write 1 is not valid). | 0 |
| 3 | JSTRT | RW0 | Injection channel conversion start state.<br>1: Injection channel conversion has started.<br>0: Injection channel conversion has not started.<br>This bit is set to 1 by hardware and cleared to 0 by software (Write 1 is not valid). | 0 |
| 2 | JEOC | RW0 | Injection into the end state of the channel group conversion.<br>1: Conversion complete.<br>0: The conversion is not completed.<br>This bit is set to 1 by hardware (All injected channels are converted) and cleared to 0 by software (Write 1 is invalid). | 0 |
| 1 | EOC | RW0 | Conversion end state.<br>1: Conversion complete.<br>0: The conversion is not completed. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | This bit is set to 1 by hardware (End of rule or injection channel group conversion), cleared by software to 0 (Write 1 is invalid) or when reading ADC_RDATAR. | |
| 0 | AWD | RW0 | Analog watchdog flag bit.<br>1: Occurrence of simulated watchdog events.<br>0: No simulated watchdog event occurred.<br>This bit is set to 1 by hardware (Conversion value is out of range of ADC_WDHTR and ADC_WDLTR registers) and cleared to 0 by software (Write 1 is not valid). | 0 |

### 8.3.2 ADC Control Register 1 (ADC_CTLR1)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MULT_CMPEN | AWD_DNFIE | AWD_OVFIE | Reserved | | BUFEN | Reserved | | AWDEN | JAWDEN | Reserved | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DISCNUM[2:0] | | | JDISCEN | DISCEN | JAUTO | AWDSGL | SCAN | JEOCIE | AWDIE | EOCIE | AWDCH[4:0] | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31 | MULT_CMPEN | RW | Multithreshold comparison function enable bits:<br>1: Enable multithreshold comparison function;<br>0: Disable multithreshold comparison function. | 0 |
| 30 | AWD_DNFIE | RW | Analog watchdog underflow interrupt enable:<br>1: Enable the analog watchdog underflow interrupt;<br>0: Disable the analog watchdog underflow interrupt. | 0 |
| 29 | AWD_OVFIE | RW | Analog watchdog overflow interrupt enable:<br>1: Enable the analog watchdog overflow interrupt;<br>0: Disable the analog watchdog overflow interrupt. | 0 |
| [28:27] | Reserved | RO | Reserved | 0 |
| 26 | BUFEN | RW | Enable BUF of ADC:<br>1: Enable;<br>0: Disable. | 0 |
| [25:24] | Reserved | RO | Reserved | 0 |
| 23 | AWDEN | RW | Analog watchdog function enable bit on the rule channel.<br>1: Enable the analog watchdog on the rule channel.<br>0: Disable the analog watchdog on the rule channel. | 0 |
| 22 | JAWDEN | RW | Analog watchdog function enable bit on the injection channel.<br>1: Enable the analog watchdog on the injection channel.<br>0: Disable the analog watchdog on the injection channel. | 0 |
| [21:16] | Reserved | RO | Reserved. | 0 |
| [15:13] | DISCNUM[2:0] | RW | Number of rule channels to be converted after external triggering in intermittent mode.<br>000: 1 channel.<br>...<br>111: 8 channels. | 0 |
| 12 | JDISCEN | RW | Inject the intermittent mode enable bit on the channel.<br>1: Enable intermittent mode on the injection channel.<br>0: Disable intermittent mode on the injection channel. | 0 |
| 11 | DISCEN | RW | Intermittent mode enable bit on rule channel. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Enable intermittent mode on the rule channel.<br>0: Disable intermittent mode on the rule channel. | |
| 10 | JAUTO | RW | After the opening of the rule channel is completed, the injection channel group enable bit is automatically switched.<br>1: Enable automatic injection channel group switching.<br>0: Disable automatic injection channel group conversion.<br>*Note: This mode requires disabling the external trigger function of the injection channel.* | 0 |
| 9 | AWDSGL | RW | In scan mode, use the analog watchdog enable bit on a single channel.<br>1: Use an analog watchdog on a single channel (AWDCH[4:0] selection).<br>0: Use analog watchdog on all channels. | 0 |
| 8 | SCAN | RW | Scan mode enable bit.<br>1: Enable scan mode (Continuous conversion of all channels selected by ADC_IOFRx and ADC_RSQRx).<br>0: Disable scan mode. | 0 |
| 7 | JEOCIE | RW | Inject the channel group end-of-conversion interrupt enable bit.<br>1: Enable the injection of the channel group conversion completion interrupt (JEOC flag).<br>0: Disable the injection channel group conversion completion interrupt. | 0 |
| 6 | AWDIE | RW | Analog watchdog interrupt enable bit.<br>1: Enable the analog watchdog interrupt.<br>0: Disable the analog watchdog interrupt.<br>*Note: In scan mode, this interrupt will abort the scan if it occurs.* | 0 |
| 5 | EOCIE | RW | End of conversion (Rule or injection channel group) interrupt enable bit.<br>1: Enable the end-of-conversion interrupt (EOC flag).<br>0: Disable the end-of-conversion interrupt. | 0 |
| [4:0] | AWDCH[3:0] | RW | Analog watchdog channel selection bits.<br>00000: Analog input channel 0.<br>00001: Analog input channel 1.<br>...<br>10011: Analog input channel 19. | 0 |

### 8.3.3 ADC Control Register 2 (ADC_CTLR2)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | SWSTART | JSWSTART | EXTTRIG | EXTSEL[3:0] | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| JEXTTRIG | JEXTSEL[3:0] | | | | ALIGN | Reserved | DMA | Reserved | | | | RSTCAL | CAL | CONT | ADON |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:23] | Reserved | RO | Reserved. | 0 |
| 22 | SWSTART | RW | To start a rule channel conversion, you need to set the software trigger to.<br>1: Initiate rule channel conversion. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0: Reset state.<br>This bit is set by software and cleared to 0 by hardware when conversion starts. | |
| 21 | JSWSTART | RW | To initiate an injection channel transition, set the software to trigger:<br>1: Initiates an injection channel transition;<br>0: Reset state.<br>This bit is set by software and is cleared to 0 by hardware or 0 by software when the conversion starts. | 0 |
| 20 | EXTTRIG | RW | Externally triggered conversion mode enable for rule channels:<br>1: Initiation of the transition using an external event;<br>0: External event start function disabled. | 0 |
| [19:16] | EXTSEL[3:0] | RW | External trigger event selection for initiating rule channel conversion.<br>0000: SWSTART software trigger;<br>0001: CC4 event of timer 1;<br>0010: CC5 event of timer 1;<br>0011: CC1 event of timer 2;<br>0100: CC2 event of timer 2;<br>0101: CC3 event of timer 2;<br>0110: CC4 event of timer 2;<br>0111: CC1 event of timer 3;<br>1000: CC2 event of timer 3;<br>1001: PA14/PB6 event;<br>1010: CC4 event/CC5 event of timer 1;<br>Other: Reserved. | 0 |
| 15 | JEXTTRIG | RW | Externally triggered conversion mode enable for the injected channels:<br>1: Initiation of the transition using an external event;<br>0: External event start function disabled. | 0 |
| [14:11] | JEXTSEL[3:0] | RW | Selection of external trigger events to initiate the injection of channel transitions:<br>0000: SWSTART software trigger;<br>0001: CC4 event of timer 1;<br>0010: CC5 event of timer 1;<br>0011: CC1 event of timer 2;<br>0100: CC2 event of timer 2;<br>0101: CC3 event of timer 2;<br>0110: CC4 event of timer 2;<br>0111: CC1 event of timer 3;<br>1000: CC2 event of timer 3;<br>1001: CC4 event/CC5 event of timer 1;<br>1010: PA14/PB6 event;<br>Other: Reserved. | 0 |
| 10 | ALIGN | RW | Data alignment.<br>1: Left-aligned; 0: Right-aligned. | 0 |
| 9 | Reserved | RO | Reserved. | 0 |
| 8 | DMA | RW | Direct Memory Access (DMA) mode enable.<br>1: Enable DMA mode.<br>0: Disable DMA mode. | 0 |
| [7:4] | Reserved | RO | Reserved. | 0 |
| 3 | RSTCAL | RW | Reset calibration, this bit is set by software, and after reset is completed, the hardware clears 0:<br>1: Initialize the calibration register;<br>0: The calibration register has been initialized.<br>*Note: If RSTCAL is being set while the conversion is being* | 0 |

| | | | | |
|---|---|---|---|---|
| | | | *performed, it takes an additional cycle to clear the calibration register.* | |
| 2 | CAL | RW | A/D calibration, this bit is set by software, and after reset is completed, the hardware clears 0:<br>1: Calibration begins;<br>0: Calibration is completed. | 0 |
| 1 | CONT | RW | Continuous conversion enable:<br>1: Continuous conversion mode;<br>0: Single conversion mode.<br>If this bit is set, conversion will be continuous until the bit is cleared. | 0 |
| 0 | ADON | RW | On/off A/D converter<br>When this bit is 0, writing 1 will wake up the ADC from power-down mode; when this bit is 1, writing 1 will start the conversion.<br>1: Turn on the ADC and start the conversion.<br>0: Turn off ADC conversion and enter power-down mode.<br>*Note: A conversion is initiated when only ADON is changed in the register, and no new conversion is initiated if there are any other bits sent for change.* | 0 |

### 8.3.4 ADC Sample Time Configuration Register 1 (ADC_SAMPTR1)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | SMP19[1:0] | | SMP18[1:0] | | SMP17[1:0] | | SMP16[1:0] | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:8] | Reserved | RO | Reserved. | 0 |
| [7:0] | SMPx[1:0] | RW | SMPx[1:0]: x=16-19, sampling time configuration for channel x:<br>00: 5.5 cycles; 01: 11.5 cycles;<br>10: 23.5 cycles; 11: 59.5 cycles;<br>These bits are used to select the sampling time for each channel independently, and the channel configuration value must remain constant during the sampling cycle. | 0 |

### 8.3.5 ADC Sample Time Configuration Register 2 (ADC_SAMPTR2)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMP15[1:0] | | SMP14[1:0] | | SMP13[1:0] | | SMP12[1:0] | | SMP11[1:0] | | SMP10[1:0] | | SMP9[1:0] | | SMP8[1:0] | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SMP7[1:0] | | SMP6[1:0] | | SMP5[1:0] | | SMP4[1:0] | | SMP3[1:0] | | SMP2[1:0] | | SMP1[1:0] | | SMP0[1:0] | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | SMPx[1:0] | RW | SMPx[1:0]: x=0-15, sampling time configuration for channel x:<br>00: 5.5 cycles; 01: 11.5 cycles; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 10: 23.5 cycles; 11: 59.5 cycles;<br>These bits are used to select the sampling time for each channel independently, and the channel configuration value must remain constant during the sampling cycle. | |

### 8.3.6 ADC Injected Channel Data Offset Register x (ADC_IOFRx) (x=1/2/3/4)

Offset address: 0x14 + (x-1)*4

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | JOFFSETx[11:0] | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | Reserved | RO | Reserved. | 0 |
| [11:0] | JOFFSETx[11:0] | RW | The data offset value of the injected channel x.<br>When converting the injected channels, this value defines the value used to subtract from the original conversion data. The result of the conversion can be read out in the ADC_IDATARx register. | 0 |

### 8.3.7 ADC Watchdog High Threshold Register (ADC_WDHTR)

Offset address: 0x24

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | HT[11:0] | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | Reserved | RO | Reserved. | 0 |
| [11:0] | HT[11:0] | RW | Analog watchdog high threshold setting value. | 0xFFF |

*Note: You can change the values of WDHTR and WDLTR during the conversion process, but they will take effect at the next conversion.*

### 8.3.8 ADC Watchdog Low Threshold Register (ADC_WDLTR)

Offset address: 0x28

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | LT[11:0] | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | Reserved | RO | Reserved. | 0 |
| [11:0] | LT[11:0] | RW | Analog watchdog low threshold setting value. | 0 |

*Note: You can change the values of WDHTR and WDLTR during the conversion process, but they will take effect at*

*the next conversion.*

### 8.3.9 ADC Regular Sequence Register 1(ADC_RSQR1)

Offset address: 0x2C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | L[3:0] | | | | SQ16[4:1] | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQ16[0] | SQ15[4:0] | | | | | SQ14[4:0] | | | | | SQ13[4:0] | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved. | 0 |
| [23:20] | L[3:0] | RW | Number of channels to be converted in a regular channel conversion sequence.<br>0000-1111: 1-16 conversions. | 0 |
| [19:15] | SQ16[4:0] | RW | The number of the 16th conversion channel in the rule sequence (0-19). | 0 |
| [14:10] | SQ15[4:0] | RW | The number of the 15th conversion channel in the rule sequence (0-19). | 0 |
| [9:5] | SQ14[4:0] | RW | The number of the 14th conversion channel in the rule sequence (0-19). | 0 |
| [4:0] | SQ13[4:0] | RW | The number of the 13th conversion channel in the rule sequence (0-19). | 0 |

### 8.3.10 ADC Regular Sequence Register 2 (ADC_RSQR2)

Offset address: 0x30

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | SQ12[4:0] | | | | | SQ11[4:0] | | | | | SQ10[4:1] | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQ10[0] | SQ9[4:0] | | | | | SQ8[4:0] | | | | | SQ7[4:0] | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:30] | Reserved | RO | Reserved. | 0 |
| [29:25] | SQ12[4:0] | RW | The number of the 12th conversion channel in the rule sequence (0-19). | 0 |
| [24:20] | SQ11[4:0] | RW | The number of the 11th conversion channel in the rule sequence (0-19). | 0 |
| [19:15] | SQ10[4:0] | RW | The number of the 10th conversion channel in the rule sequence (0-19). | 0 |
| [14:10] | SQ9[4:0] | RW | The number of the 9th conversion channel in the rule sequence (0-19). | 0 |
| [9:5] | SQ8[4:0] | RW | The number of the 8th conversion channel in the rule sequence (0-19). | 0 |
| [4:0] | SQ7[4:0] | RW | The number of the 7th conversion channel in the rule sequence (0-19). | 0 |

### 8.3.11 ADC Regular Sequence Register 3 (ADC_RSQR3)

Offset address: 0x34

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | SQ6[4:0] | | | | | SQ5[4:0] | | | | | SQ4[4:1] | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SQ4[0] | | SQ3[4:0] | | | | | SQ2[4:0] | | | | | SQ1[4:0] | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:30] | Reserved | RO | Reserved. | 0 |
| [29:25] | SQ6[4:0] | RW | The number of the 6th conversion channel in the rule sequence (0-19). | 0 |
| [24:20] | SQ5[4:0] | RW | The number of the 5th conversion channel in the rule sequence (0-19). | 0 |
| [19:15] | SQ4[4:0] | RW | The number of the 4th conversion channel in the rule sequence (0-19). | 0 |
| [14:10] | SQ3[4:0] | RW | The number of the 3rd conversion channel in the rule sequence (0-19). | 0 |
| [9:5] | SQ2[4:0] | RW | The number of the 2nd conversion channel in the rule sequence (0-19). | 0 |
| [4:0] | SQ1[4:0] | RW | The number of the 1st conversion channel in the rule sequence (0-19). | 0 |

### 8.3.12 ADC Injected Sequence Register (ADC_ISQR)

Offset address: 0x38

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | JL[1:0] | | JSQ4[4:1] | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| JSQ4[0] | | JSQ3[4:0] | | | | | JSQ2[4:0] | | | | | JSQ1[4:0] | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:22] | Reserved | RO | Reserved. | 0 |
| [21:20] | JL[1:0] | RW | Inject the number of channels to be converted in the channel conversion sequence.<br>00-11: 1-4 conversions. | 0 |
| [19:15] | JSQ4[4:0] | RW | The number of the 4th conversion channel in the injection sequence (0-19).<br>*Note: The software writes and assigns the channel number (0-19) as the 4th in the sequence to be converted.* | 0 |
| [14:10] | JSQ3[4:0] | RW | The number of the 3rd conversion channel in the injection sequence (0-19). | 0 |
| [9:5] | JSQ2[4:0] | RW | The number of the 2nd conversion channel in the injection sequence (0-19). | 0 |
| [4:0] | JSQ1[4:0] | RW | The number of the 1st conversion channel in the injection sequence (0-19). | 0 |

*Note: Unlike the regular conversion sequence, if the length of JL[1:0] is less than 4, the sequence order of conversion starts from (4 - JL).*

*For example, when JL[1:0]=3 (4 injected transitions in the sequencer), the ADC will convert channels in the following order: JSQ1[4:0], JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0];*

*When JL[1:0]=2 (3 injected transitions in the sequencer), the ADC will convert the channels in the following order: JSQ2[4:0], JSQ3[4:0] and JSQ4[4:0];*
*When JL[1:0]=1 (2 injected conversions in the sequencer), the ADC converts the channels in the following order: first JSQ3[4:0], then JSQ4[4:0];*
*When JL[1:0] = 0 (1 injection conversion in the sequencer), the ADC will convert only the JSQ4[4:0] channels.*
*If ADCx_ISQR[21:0]=10 00111 00011 00111 00010, the ADC will convert channels in the following order: JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0], indicating that the scan conversions are performed in the following channel order: 7, 3, 7.*

### 8.3.13 ADC Injected Data Register (ADC_IDATARx) (x=1/2/3/4)

Offset address: 0x3C + (x-1)*4

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| JDATA[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0 |
| [15:0] | JDATA[15:0] | RO | Injection of channel conversion data (Data left-aligned or right-aligned). | 0 |

### 8.3.14 ADC Regular Data Register (ADC_RDATAR)

Offset address: 0x4C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | REGU_CH[3:0] | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:20] | Reserved | RO | Reserved. | 0 |
| [19:16] | REGU_CH[3:0] | RO | When the channel indication function is turned on, it indicates the order of the conversion channel in the current conversion data corresponding to the rule sequence. | 0 |
| [15:0] | DATA[15:0] | RO | Rule channel conversion data (Data left-aligned or right-aligned) | 0 |

### 8.3.15 ADC Control Register 3 (ADC_CTLR3)

Offset address: 0x50

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | RDR_POLL | CMP_XOR_MODE | AVG_DNFIE | AVG_OVFIE | AVG_EN | AVG_K[2:0] | | | DNF | OVF | DNFIF | OVFIF |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | AWD2_RES | AWD1_RES | AWD0_RES | CH_SAVE | AWD2_RST_EN | AWD1_RST_EN | AWD0_RST_EN | AWD_SCAN | Reserved | | SW_DIS |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:28] | Reserved | RO | Reserved | 0 |
| 27 | RDR_POLL | RW | Rule channel data polling save function enable:<br>1: The rule channel polling function is enabled, and the results of rule channel conversion are stored in ADC_RDATAR, ADC_RDATAR2, ADC_RDATAR3, and ADC_RDATA4 in sequence;<br>0: The rule channel polling function is disabled, and the conversion results are stored in ADC_RDATAR. | 0 |
| 26 | CMP_XOR_MODE | RW | Comparison result XOR mode selection:<br>1: The comparison result is XOR at the end of each channel conversion;<br>0: The comparison result is XOR at the end of a 4-channel polling comparison. | 0 |
| 25 | AVG_DNFIE | RW | Conversion underflow interrupt is enabled when open the sliding average function:<br>1: Enable;<br>0: Disable. | 0 |
| 24 | AVG_OVFIE | RW | Conversion overflow interrupt is enabled when open the sliding average function:<br>1: Enable;<br>0: Disable. | 0 |
| 23 | AVG_EN | RW | Sliding average function enable:<br>1: Enable;<br>0: Disable. | 0 |
| [22:20] | AVG_K[2:0] | RW | Sliding average window value settings:<br>000: 32 value average;<br>001: 64 value average;<br>010: 128 value average;<br>011: 256 value average;<br>100: 512 value average;<br>Other: 1024 value average. | 0 |
| 19 | DNF | RO | Conversion underflow flag when open the sliding average function:<br>1: The conversion value is less than the average value minus the low threshold configuration value;<br>0: The conversion value is greater than or equal to the average value minus the low threshold configuration value. | 0 |
| 18 | OVF | RO | Conversion overflow flag when open the sliding average function:<br>1: The conversion value is greater than the average value plus the high threshold configuration value;<br>0: The conversion value is less than or equal to the average value plus the high threshold configuration value. | 0 |
| 17 | DNFIF | R0 | Conversion underflow interrupt flag when open the sliding average function (edge interrupt).<br>1: The conversion value is less than the average value minus the low threshold configuration value;<br>0: The conversion value is greater than or equal to the | 0 |

| | | | | |
|---|---|---|---|---|
| | | | average value minus the low threshold configuration value. | |
| 16 | OVFIF | R0 | Conversion overflow interrupt flag when open the sliding average function (edge interrupt).<br>1: The conversion value is greater than the average value plus the high threshold configuration value;<br>0: The conversion value is less than or equal to the average value plus the high threshold configuration value. | 0 |
| [15:11] | Reserved | RO | Reserved | 0 |
| 10 | AWD2_RES | RW0 | Analog watchdog 2 comparison results.<br>1: The conversion value is greater than the watchdog 2 high threshold or less than the watchdog 2 low threshold;<br>0: the conversion value is between the watchdog 2 high threshold and low threshold.<br>*Note: Hardware set to 1, write 0 to clear.* | 0 |
| 9 | AWD1_RES | RW0 | Analog watchdog 1 comparison results.<br>1: The conversion value is greater than the watchdog 1 high threshold or less than the watchdog 1 low threshold;<br>0: the conversion value is between the watchdog 1 high threshold and low threshold.<br>*Note: Hardware set to 1, write 0 to clear.* | 0 |
| 8 | AWD0_RES | RW0 | Analog watchdog 0 comparison results.<br>1: The conversion value is greater than the watchdog 0 high threshold or less than the watchdog 0 low threshold;<br>0: the conversion value is between the watchdog 0 high threshold and low threshold.<br>*Note: Hardware set to 1, write 0 to clear.* | 0 |
| 7 | CH_SAVE | RW | Rule data channel indication function enable bit:<br>1: Enable the channel indication function, ADC_RDATARx contains channel information;<br>0: Disable the channel indication function, ADC_RDATARx is only the conversion result. | 0 |
| 6 | AWD2_RST_EN | RW | Analog watchdog 2 reset enable bit:<br>1: On<br>0: Off. | 0 |
| 5 | AWD1_RST_EN | RW | Analog watchdog 1 reset enable bit:<br>1: On<br>0: Off. | 0 |
| 4 | AWD0_RST_EN | RW | Analog watchdog 0 reset enable bit:<br>1: On<br>0: Off. | 0 |
| 3 | AWD_SCAN | RW | Analog watchdog scan enable:<br>1: Turn on watchdog scanning;<br>0: Turn off watchdog scanning. | 0 |
| [2:1] | Reserved | RO | Reserved | 0 |
| 0 | SW_DIS | RW | ADC channel advance switching function disable control bit:<br>1: Disable the channel advance switching function, i.e. switch to the next channel after the current conversion is finished;<br>0: Enable channel advance switching function, i.e., switch to the next channel after the end of the current channel sampling. | 0 |

### 8.3.16 ADC Watchdog 1 Threshold Register (ADC_WDTR1)

Offset address: 0x54

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| Reserved | HTR1 |
|---|---|

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

| Reserved | LTR1 |
|---|---|

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:28] | Reserved | RO | Reserved | 0 |
| [27:16] | HTR1 | RW | Analog watchdog high threshold setting value. | 0xFFF |
| [15:12] | Reserved | RO | Reserved | 0 |
| [11:0] | LTR1 | RW | Analog watchdog low threshold setting value. | 0 |

*Note: Only available for watchdog channel 1.*

### 8.3.17 ADC Watchdog 2 Threshold Register (ADC_WDTR2)

Offset address: 0x58

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

| Reserved | HTR2[11:0] |
|---|---|

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

| Reserved | LTR2[11:0] |
|---|---|

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:28] | Reserved | RO | Reserved. | 0 |
| [27:16] | HTR2 | RW | Analog watchdog high threshold setting value. | 0xFFF |
| [15:12] | Reserved | RO | Reserved | 0 |
| [11:0] | LTR2 | RW | Analog watchdog low threshold setting value. | 0 |

*Note: Only available for watchdog channel 2.*

### 8.3.18 ADC Watchdog 3 Threshold Register (ADC_WDTR3)

Offset address: 0x5C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

| Reserved | HTR3[11:0] |
|---|---|

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

| Reserved | LTR3[11:0] |
|---|---|

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0 |
| [27:16] | HTR3 | RW | Analog watchdog high threshold setting value. | 0xFFF |
| [15:12] | Reserved | RO | Reserved | 0 |
| [11:0] | LTR3 | RW | Analog watchdog low threshold setting value. | 0 |

*Note: The limit conditions for determining the conversion value underflow/overflow are: Sliding average + setting the high threshold of the average value is less than or equal to 4095, and sliding average - setting the low threshold of the average value is greater than or equal to 0.*

### 8.3.19 ADC Delay Data Register (ADC_DLYR)

Offset address: 0x60

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DLYSRC | DLYVLU[8:0] | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:10] | Reserved | RO | Reserved | 0 |
| 9 | DLYSRC | RW | External trigger source delay selection.<br>1: Injection channel external trigger delay;<br>0: Rule channel external trigger delay. | 0 |
| [8:0] | DLYVLU[8:0] | RW | External trigger delay data, delay time configuration, unit: ADC clock cycle. | 0 |

### 8.3.20 ADC Average Data Register (ADC_DLYR)

Offset address: 0x64

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | AVG_DR[11:0] | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | Reserved | RO | Reserved | 0 |
| [11:0] | AVG_DR | RO | ADC sliding average value. | 0 |

### 8.3.21 ADC Rule Data Register x (ADC_RDATARx) (x=2/3/4)

Offset address: 0x68+ (x-2)*4

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | DATA[11:0] | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:12] | Reserved | RO | Reserved | 0 |
| [11:0] | DATA[11:0] | RO | When the rule channel data polling and saving is enabled, the rule channel switches 4 to 1 cycles, and the xth channel in the cycle corresponds to the converted data. | 0 |

# Chapter 9 Advanced-control Timer (ADTM)

The Advanced-control timer Module contains one powerful 16-bit auto-reload timers (TIM1), which can be used to measure pulse width or generate pulses, PWM waves, etc. It is used in motor control, power supply, etc.

## 9.1 Main Features

The main features of the advanced-control timer (TIM1) include.

- 16-bit auto-reload counter supporting incremental counting mode, decremental counting mode and incremental and decremental counting mode.
- 16-bit prescaler with dynamically adjustable crossover coefficients from 1 to 65536.
- Support 4 independent comparison capture channels.
- Each comparison capture channel supports multiple operating modes, such as: input capture, output comparison, PWM generation and single pulse output.
- Support complementary outputs with programmable dead time;
- Support external signals to control the timer.
- Support updating the timer after a defined period using a repeat counter.
- Support resetting the timer or placing it in the OK state using the brake signal.
- Support the use of DMA in multiple modes.
- Support incremental encoders.
- Support cascading and synchronization between timers.

## 9.2 Principle and Structure

This section deals with the internal construction of advanced-control timers.

### 9.2.1 Overview

As shown in Figure 9-1, the structure of the advanced-control timer can be roughly divided into 3 parts, namely the input clock part, the core counter part and the compare capture channel part.

The advanced-control timer can be clocked from the HB bus clock (CK_INT), from an external clock input pin (TIMx_ETR), from other timers with clock output (ITRx), or from the input of the compare capture channel (TIMx_CHx). These input clock signals become the CK_PSC clock after various set filtering and dividing operations and are output to the core counter section. In addition, these complex clock sources can also be output as TRGO to other peripherals such as timers and ADCs.

The core of the advanced timer is a 16-bit counter (CNT). After frequency division by prescaler (PSC), CK_PSC becomes CK_CNT and is output to CNT. CNT supports counting up mode, counting down mode and counting up and down mode, and there is an automatic reloading value register (ATRLR) to reload the initial value for CNT after each counting period. In addition, an auxiliary counter counts the number of times ATRLR is the initial value of CNT reloading. When the number of times reaches the number set in the repeat count register (RPTCR), a specific event can be generated.

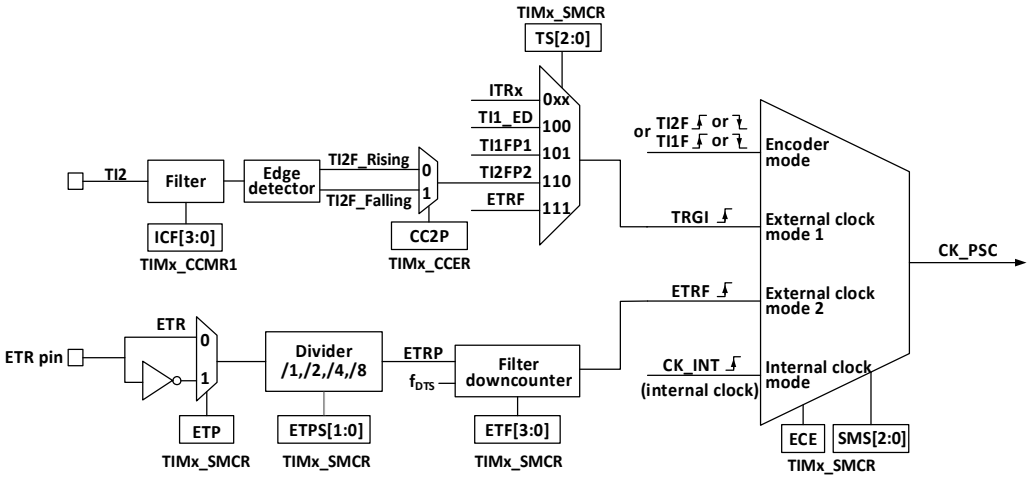
The advanced timer has 4 groups of comparison capture channels, and each group of compare capture channels can input pulses from its own pins or output waveforms to the pins, that is, the compare capture channels support input

and output modes. The input of each channel of the compare capture register supports filtering, frequency division, edge detection and other operations, and supports mutual triggering between channels, and can also provide a clock for the core counter CNT. Each comparison capture channel has a set of compare capture registers (CHxCVR), which support comparison with the main counter (CNT) and output pulses.

Figure 9-1 Block diagram of advanced-control timer structure

## 9.2.2 Clock Input

Figure 9-2 Block diagram of CK_PSC source for advanced-control timer




The advanced-control timer CK_PSC has many clock sources and can be divided into 4 categories.

1) Route of external clock pin (ETR) input clock: ETR → ETRP → ETRF.
2) Internal HB clock input route: CK_INT.
3) Route from the comparison capture channel pin (TIM1_CHx): TIM1_CHx → TIx → TIxFPx, this route is also used in encoder mode.
4) Inputs from other internal timers: ITRx.

The actual operation can be divided into 4 categories by determining the choice of input pulse for the SMS of the CK_PSC source.

1) Selection of the internal clock source (CK_INT).
2) External clock source mode 1.
3) External clock source mode 2.
4) Encoder mode.

All 4 clock source sources mentioned above can be selected by these 4 operations.

### 9.2.2.1 Internal Clock Source (CK_INT)

If the SMS field is held at 000b to start the advanced-control timer, then it is the internal clock source (CK_INT) that is selected as the clock. At this point CK_INT is CK_PSC.

### 9.2.2.2 External Clock Source Mode 1

When the SMS domain is set to 111b, external clock source mode 1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source of CK_PSC. it is worth noting that the source of TRGI also needs to be selected by configuring the TS domain. the TS domain can select the following types of pulses as clock sources.

1) Internal trigger (ITRx, x is 0,1,2,3).
2) Comparison of the signal after capturing channel 1 through the edge detector (TI1F_ED).
3) Comparison of signals TI1FP1, TI2FP2 of the capture channel.
4) The signal ETRF from the external clock pin input.

#### 9.2.2.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE position is set, the external clock source mode 2 is used. when using the external clock source mode 2, ETRF is selected as CK_PSC. the ETR pin becomes ETRP after passing through the optional inverter (ETP), divider (ETPS), and then ETRF after passing through the filter (ETF).

With the ECE position bit and the SMS set to 111b, this is equivalent to the TS selecting ETRF as an input.

#### 9.2.2.4 Encoder Mode

Setting the SMS to 001b, 010b, 011b will enable the encoder mode. Enabling encoder mode allows you to select a specific level in TI1FP1 and TI2FP2 to signal the output with another jump edge as the signal. This mode is used when an external encoder is used. Refer to Section 9.3.10 for specific functions.

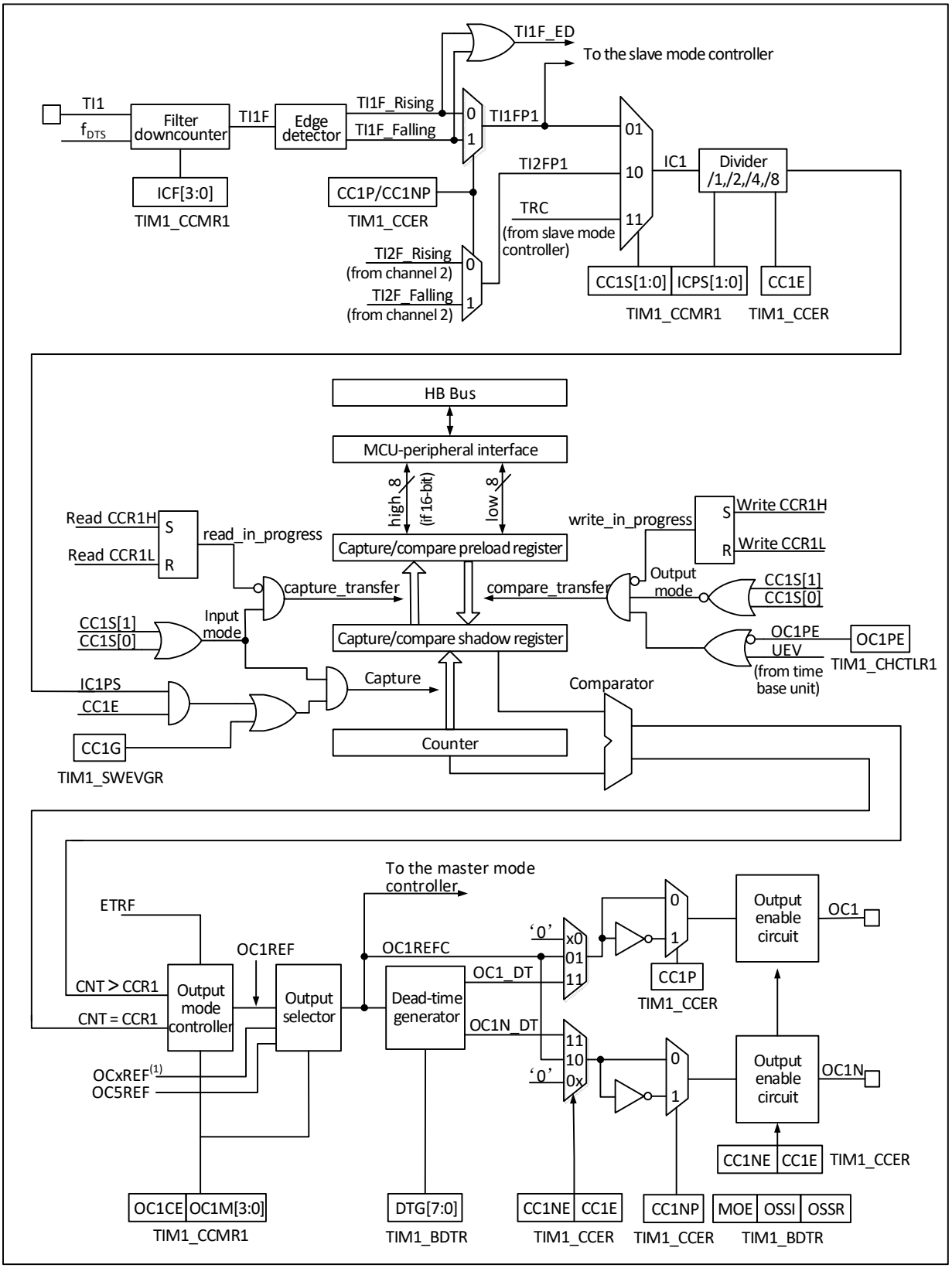
### 9.2.3 Counters and Peripherals

CK_PSC is input to the prescaler (PSC) for dividing. the PSC is 16-bit and the actual dividing factor is equal to the value of R16_TIMx_PSC + 1. CK_PSC goes through the PSC and becomes CK_INT. changing the value of R16_TIM1_PSC does not take effect in real time, but is updated to the PSC after an update event. the update event includes a UG bit clear and reset. The core of the timer is a 16-bit counter (CNT). CK_CNT is eventually fed to the CNT, which supports incremental count mode, decremental count mode, and incremental and decremental count modes, and has an Automatic Reload Register (ATRLR) that reloads the initial value for the CNT at the end of each count cycle. There is also an auxiliary counter that keeps track of the number of times the ATRLR reloads the initial value for the CNT and can generate a specific event when the number of times set in the Repeat Count Register (RPTCR) is reached.

### 9.2.4 Compare/Capture Channels and Peripherals

The core of the timer is the compare/capture register, which is complemented by digital filtering, frequency division and inter-channel alternate in the peripheral input section, comparator and output control in the output section.

Figure 9-3 Block diagram of the structure of the compare/capture channel




The structure block diagram of the compare/capture channel is shown in Figure 9-3. The signal is input from the channel x pin and optionally made as TIx (the source of TI1 can be more than just CH1, see the structure block diagram of timer 10-1), TI1 is passed through the filter (ICF[3:0]) to generate TI1F, and then divided into

TI1F_Rising and TI1F_Falling through the edge detector, these two signals are selected (CC1P) to generate TI1FP1, TI1FP1 and TI2FP1 from channel 2 are sent together to CC1S to select to become IC1, which is sent to the comparison capture register after ICPS dividing.

The compare capture register consists of a preload register and a shadow register, and the read/write process operates only on the preload register. In capture mode, the capture occurs on the shadow register and is then copied to the preload register; in compare mode, the contents of the preload register are copied to the shadow register, and then the contents of the shadow register are compared to the core counter (CNT).

## 9.3 Function and Implementation

The implementation of the complex functions of the advanced-control timer are all achieved by the operation of the timer's compare capture channel, clock input circuit and counter and peripheral parts. The clock input to the timer can come from multiple clock sources, including the input to the compare capture channel. The operation of the compare capture channel and clock source selection directly determines its function. The compare capture channel is bidirectional and can operate in both input and output modes.

### 9.3.1 Input Capture Mode

The input capture mode is one of the basic functions of the timer. The principle of input capture mode is that a capture event occurs when a determined edge on the ICxPS signal is detected, and the current value of the counter is latched into the compare capture register (R16_TIM1_CHCTLRx). When a capture event occurs, CCxIF (in R16_TIM1_INTFR) is set, and if an interrupt or DMA is enabled, the corresponding interrupt or DMA is also generated. if CCxIF is already set when a capture event occurs, the CCxOF bit is set. CCxIF can be cleared by software, or by hardware by reading the compare capture register. CCxOF is cleared by software.

An example of channel 1 to illustrate the steps to use the input capture mode is as follows:

1) Configure the CCxS domain to select the source of the ICx signal. For example, set to 10b and select TI1FP1 as the source of IC1 instead of using the default setting, where the CCxS domain defaults to making the comparison capture module the output channel.
2) Configure the ICxF domain to set the digital filter for the TI signal. The digital filter will sample the signal at a determined frequency, a determined number of times, and then output a hop. This sampling frequency and number of times is determined by ICxF.
3) Configure the CCxP bit to set the polarity of the TIxFPx. For example, keeping the CC1P bit low and selecting rising edge jumps.
4) Configure the ICxPS domain to set the ICx signal to be the crossover factor between ICxPS. For example, keeping ICxPS at 00b, without crossover.
5) Configure the CCxE bit to allow capturing the value of the core counter (CNT) into the compare capture register. Set the CC1E bit.
6) Configure the CCxIE and CCxDE bits as needed to determine whether to allow enable interrupts or DMA.

This completes the comparison capture channel configuration.

When a captured pulse is input to TI1, the value of the core counter (CNT) is recorded in the compare capture register, CC1IF is set, and the CCIOF bit is set when CC1IF has been set before. If the CC1IE bit is set, then an interrupt is generated; if CC1DE is set, a DMA request is generated. An input capture event can be generated by software by writing the event generation register (TIM1_SWEVGR).

### 9.3.2 Compare Output Mode

The compare output mode is one of the basic functions of a timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) agrees with the value of the compare capture register. the OCxM field (In R16_TIM1_CHCTLRx) and the CCxP bit (in R16_TIM1_CCER) determine whether the output is a definite high or low level or a level flip. The CCxIF bit is also set when a compare coherent event is generated, an interrupt is generated if the CCxIE bit is pre-set, and a DMA request is generated if the CCxDE bit is pre-set.

The steps to configure to compare output mode are as follows:

1) Configure the clock source and auto-reload value of the core counter (CNT);
2) Set the count value to be compared to the compare capture register (R16_TIM1_CHxCVR);
3) setting the CCxIE bit if an interrupt needs to be generated;
4) disable the preload register of the compare register by keeping OCxPE to 0;
5) set the output mode, setting the OCxM field and the CCxP bit;
6) Enable the output, setting the CCxE bit;
7) Set the CEN bit to start the timer.

### 9.3.3 Forced Output Mode

The output pattern of the timer's compare capture channel can be forced by software to output a determined level without relying on comparison of the compare capture register's shadow register with the core counter.

This is done by setting OCxM to 100b, which forces OCxREF to low, or by setting OCxM to 101b, which forces OCxREF to high.

Note that by forcing OCxM to 100b or 101b, the comparison process of the internal core counters and compare capture registers is still going on, the corresponding flags are still set, and interrupts and DMA requests are still being generated.

### 9.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of PWM and is a special case of the input capture mode. The operation is the same as input capture mode except for the following differences: PWM occupies two compare capture channels and the input polarity of the two channels is set to opposite, one of the signals is set as trigger input and SMS is set to reset mode.

For example, to measure the period and frequency of the PWM wave input from TI1, the following operations are required:

1) Set TI1 (TI1FP1) to be the input of IC1 signal. Set CC1S to 01b.
2) Set TI1FP1 to rising edge active. Holding CC1P at 0.
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b.
4) Select TI1FP2 to set to falling edge active. Set CC2P to 1.
5) Select TI1FP1 as the source of the clock source. set TS to 101b.
6) Set the SMS to reset mode, i.e. 100b.
7) Enables input capture. CC1E and CC2E are set.

Thus the value of compare capture register 1 is the period of the PWM, and the value of compare capture register 2

is its duty cycle.

*Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIM1_CH1/TIM1_CH2 can be used in the PWM input mode.*

### 9.3.5 PWM Output Mode

PWM output mode is one of the basic functions of the timer. PWM output mode is most commonly used to determine the PWM frequency using the reload value and the duty cycle using the capture comparison register. Set 110b or 111b in the OCxM field to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit to enable automatic reload of the preload register. Since the value of the preload register can only be sent to the shadow register when an update event occurs, the UG bit needs to be set to initialize all registers before the core counter starts counting. In PWM mode, the core counter and the compare capture register are always comparing, and depending on the CMS bit, the timer is able to output edge-aligned or center-aligned PWM signals.

- Edge aligned

When edge aligned is used, the core counter is incremented or decremented, and in the PWM mode 1 scenario, OCxREF is high when the core counter value is greater than the compare capture register, and low when the core counter value is less than the compare capture register (e.g., when the core counter grows to the value of R16_TIM1_ATRLR and reverts to all zeros).

- Center aligned

When using the center aligned modes, the core counter runs in alternating incremental and decremental count modes, and OCxREF makes rising and falling jumps when the values of the core counter and the compare capture register match. However, the comparison flags are set at different times in the three central alignment modes. When using the center aligned modes, it is best to generate a software update flag (Set the UG bit) before starting the core counter.

The comparison channel 5 only generates comparison events and DMA requests, and cannot output PWM through the pin. Complementary channels 1N/2N/3N can choose to output independent PWM function in edge alignment mode: the mode of output PWM can be set through OC3NM, OC2NM and OC1NM bits, and CC3N_SPEC_EN, CC2N_SPEC_EN and CC1N_SPEC_EN turn on the independent output PWM function of each corresponding complementary channel, and the comparison value corresponds to the setting values of CH8CVR, CH7CVR and CH6CVR.

Enable the CMS_SPEC bit to turn on the downward shift function in the central symmetric mode, and use different comparison values in the ascending and descending stages when moving to the central symmetric mode. The setting values of CH6CVR, CH7CVR and CH8CVR correspond to the comparison values of the descending stages of channel 1, channel 2 and channel 3 respectively.

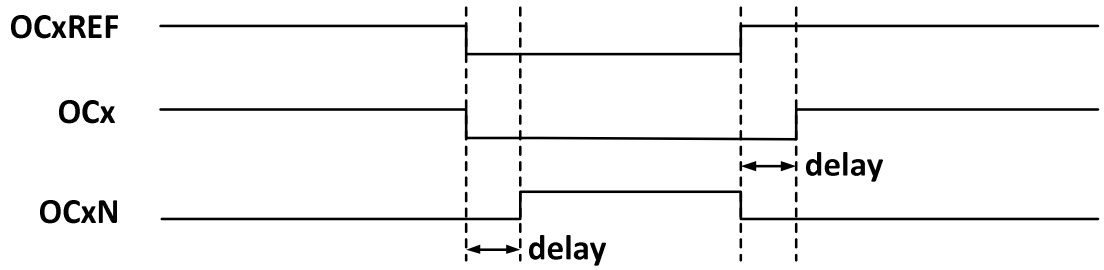
### 9.3.6 Complementary Outputs and Dead-time Insertion

The compare capture channel generally has two output pins (Compare capture channel 4 has only one output pin) and can output two complementary signals (OCx and OCxN). OCx and OCxN can be independently set for polarity via the CCxP and CCxNP bits, independently set for output enable via CCxE and CCxNE, and independently set for output enable via the MOE, OIS, OISN, OSSI, and OSSR bits for dead-time and other controls. Enabling the OCx and OCxN outputs simultaneously will insert a dead-time, and each channel has a 10-bit dead-time generator. OCx and OCxN are generated by the OCxREF association. If both OCx and OCxN are high active, then OCx is the same as OCxREF except that the rising edge of OCx is equivalent to OCxREF with a delay, and OCxN is the

opposite of OCxREF in that its rising edge will have a delay relative to the falling edge of the reference signal. If the delay is greater than the effective output width, the corresponding pulse will not be generated.

The relationship between OCx and OCxN and OCxREF is illustrated in Figure 12-4, which shows the dead-time.

Figure 9-4 Complementary outputs and dead-time insertion




### 9.3.7 Dead-time Asymmetry

On the basis of complementary output configuration and dead-time control, by configuring the DT_VLU2 register (the configuration value should be greater than 0 and less than the number of TDTGs set in the TIM1_BDTR register DTG[7:0]), the DTP_MODE bit or DTN_MODE bit is configured to select the dead-time of DT_VLU2 configuration to occur on the rising or falling edge of OCXREF. The dead time asymmetry is realized by this method.

### 9.3.8 Brake Signal

When the brake signal is generated, the output enable signal and invalid level are modified according to the MOE, OIS, OISN, OSSI, and OSSR bits. However, OCx and OCxN will not be at the active level at any time. The source of the brake event can come from the brake input pin or it can be a clock failure event which is generated by the CSS (Clock Safety System).

After system reset, the brake function is disabled by default (MOE bit is low), and setting the BKE bit enables the brake function. The polarity of the input brake signal can be set by setting BKP, and the BKE and BKP signals can be written at the same time, and there is a delay of one HB clock before the actual writing, so you need to wait for one HB cycle to read the written value correctly.

At the presence of the selected level on the brake pin the system will generate the following actions.

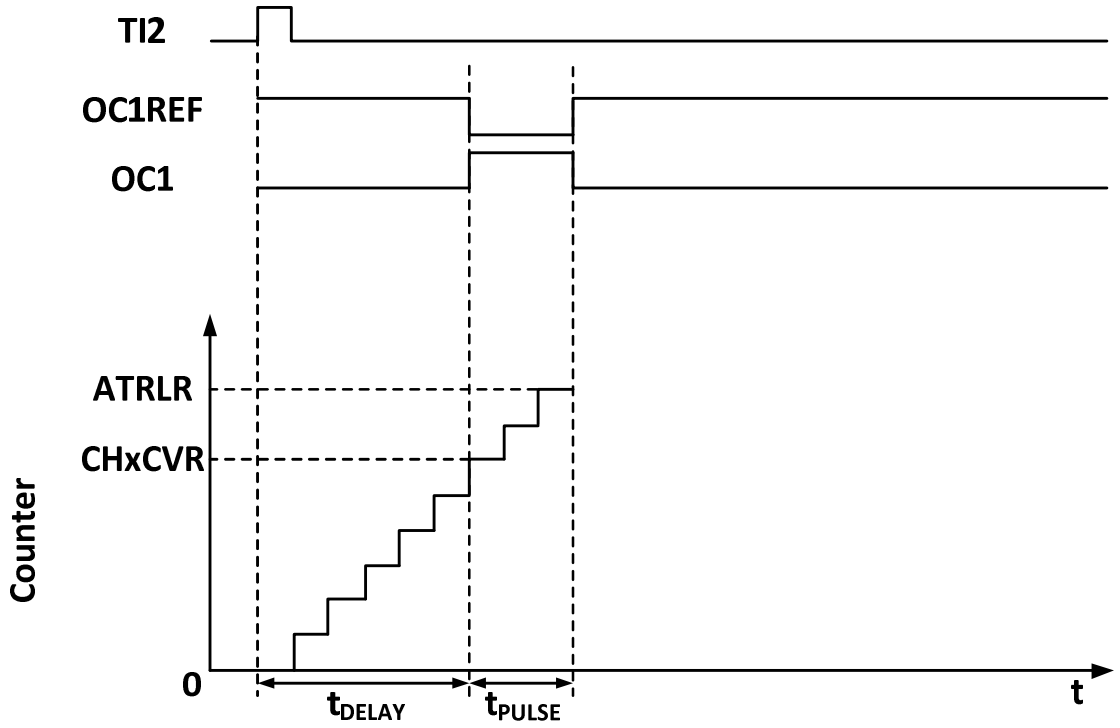
1) The MOE bit is cleared asynchronously, setting the output to an invalid, idle or reset state, depending on the setting of the SOOI bit.
2) After the MOE has been cleared, each output channel outputs a level determined by OISx.
3) When using complementary outputs: the outputs are placed in a null state, depending on the polarity.
4) If the BIE is set, an interrupt is generated when the BIF is set; if the BDE bit is set, a DMA request is generated.
5) If the AOE is set, the MOE bit is automatically set at the next update event UEV.

### 9.3.9 One-pulse Mode

One-pulse mode (OPM) can be used to allow the microcontroller to respond to a specific event by causing it to generate a pulse after a delay, with the delay and width of the pulse programmable. Placing the OPM bit allows the core counter to stop when the next update event UEV is generated (Counter flips to 0).

As shown in Figure 9-5, a positive pulse of length Tpulse needs to be generated on OC1 after a delay Tdelay at the beginning of a rising edge detected on the TI2 input pin.

Figure 9-5 Generation of one pulse mode




1) Set TI2 to trigger. Setting the CC2S field to 01b to map TI2FP2 to TI2; setting the CC2P bit to 0b to set TI2FP2 as rising edge detection; setting the TS field to 110b to set TI2FP2 as trigger source; setting the SMS field to 110b to set TI2FP2 to be used to start the counter.
2) Tdelay is determined by the value of the Compare Capture Register, and Tpulse is determined by the value of the Auto Reload Value Register and the Compare Capture Register.

### 9.3.10 Encoder Mode

The encoder mode is a typical application of the timer and can be used to access the biphasic output of the encoder. The counting direction of the core counter is synchronized with the direction of the encoder's rotation axis, and each pulse output from the encoder will cause the core counter to add or subtract one. To use the encoder, set the SMS field to 001b (Count only on TI2 edge), 010b (Count only on TI1 edge) or 011b (Count on both TI1 and TI2 edges), connect the encoder to the input of comparison capture channels 1 and 2, and set a value for the reload value register, which can be set to a larger value. When in encoder mode, the internal compare capture register, prescaler, repeat count register, etc. of the timer are working normally. The following table shows the relationship between the counting direction and the encoder signal.

Table 9-1 Relationship between counting direction and encoder signal of timer encoder mode

| Counting effective edges | The level of relative signals | TI1FP1 signal edge | | TI2FP2 signal | |
|---|---|---|---|---|---|
| | | Rising edge | Falling edge | Rising edge | Falling edge |
| Counting at TI1 edge only | high | Downward counting | Upward counting | No count | |
| | low | Upward counting | Downward counting | | |
| Counting at TI2 edge only | high | No count | | Upward counting | Downward counting |
| | low | | | Downward counting | Upward counting |
| Double edge counting at TI1 and TI2 | high | Downward counting | Upward counting | Upward counting | Downward counting |
| | low | Upward counting | Downward counting | Downward counting | Upward counting |

### 9.3.11 Timer Synchronization Mode

Timer can output clock pulse (TRGO) and also receive input from other timers (ITRx). The source of ITRx of different timers (TRGO of other timers) is different.

Table 9-2 TIM1 internal trigger connection

| Slave timer | ITR0(TS=000) | ITR1(TS=001) | ITR2(TS=010) | ITR3(TS=011) |
|---|---|---|---|---|
| TIM1 | TIM2 | | USB | |

### 9.3.12 Dual Edge Capture Mode

The pulse can be measured by turning on the double-edge capture of the corresponding channel through the CAP_ED_CHx bit in the TIM1_AUX1 register.

For example, channel 2 is used to capture the pulse width, and the source of IC2 signal (CC2S bit in TIM1_CHCTLR1 register is 11b) is selected to enable the double-edge capture function of channel 2 (CAP_ED_CH2 bit in TIM1_AUX1 register is 1). So far, the configuration of double edge capture has been completed.

The high and low level width values of the captured pulse can be read through the CH2CVR bit in the TIM1_CH2CVR register, and the level corresponding to the captured value can be indicated through the TIM1_CH2CVR register bit[16] by turning on the CAPLVL bit.

### 9.3.13 High Frequency Clock Counting Mode

Channels 1, 2, 3 and corresponding complementary channels of the advanced timer support the use of PLL clock as the working clock of the core counter to generate PWM(OCxM bit can only select PWM mode 1 or PWM mode 2), and only the incremental counting mode is supported under this function; When using the dead-time function in this mode, the TIM1_BDTR register DTG[7] is the dead-time function enabling bit, DTG[3:0] is configured at the dead-time value of the falling edge of OCXREF, DT _ VLU2 [3: 0] is configured at the dead-time value of the rising edge of OCXREF, and bit[6] of DTG can turn on the dead-time counting clock to divide by 4 and extend the dead-time function.

### 9.3.14 Debug Mode

When the system enters debug mode, the timer continues to run or stops according to the settings of the DBG module.

## 9.4 Register Description

Table 9-3 TIM1-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_TIM1_CTLR1 | 0x40012C00 | Control register 1 | 0x0000 |
| R16_TIM1_CTLR2 | 0x40012C04 | Control register 2 | 0x0000 |
| R16_TIM1_SMCFGR | 0x40012C08 | Slave mode control register | 0x0000 |
| R16_TIM1_DMAINTENR | 0x40012C0C | DMA/interrupt enable register | 0x0000 |
| R16_TIM1_INTFR | 0x40012C10 | Interrupt status register | 0x0000 |
| R16_TIM1_SWEVGR | 0x40012C14 | Event generation register | 0x0000 |
| R16_TIM1_CHCTLR1 | 0x40012C18 | Compare/capture control register 1 | 0x0000 |
| R16_TIM1_CHCTLR2 | 0x40012C1C | Compare/capture control register 2 | 0x0000 |
| R16_TIM1_CCER | 0x40012C20 | Compare/capture enable register | 0x0000 |
| R16_TIM1_CNT | 0x40012C24 | Counters | 0x0000 |
| R16_TIM1_PSC | 0x40012C28 | Counting clock prescaler | 0x0000 |
| R16_TIM1_ATRLR | 0x40012C2C | Auto-reload value register | 0xFFFF |
| R16_TIM1_RPTCR | 0x40012C30 | Repeat count Register | 0x0000 |
| R32_TIM1_CH1CVR | 0x40012C34 | Compare/capture register 1 | 0x00000000 |
| R32_TIM1_CH2CVR | 0x40012C38 | Compare/capture register 2 | 0x00000000 |
| R32_TIM1_CH3CVR | 0x40012C3C | Compare/capture register 3 | 0x00000000 |
| R32_TIM1_CH4CVR | 0x40012C40 | Compare/capture register 4 | 0x00000000 |
| R16_TIM1_BDTR | 0x40012C44 | Brake and deadtime registers | 0x0000 |
| R16_TIM1_DMACFGR | 0x40012C48 | DMA control register | 0x0000 |
| R16_TIM1_DMAADR | 0x40012C4C | DMA address register in continuous mode | 0x0000 |
| R16_TIM1_AUX1 | 0x40012C50 | Auxiliary configuration register 1 | 0x0000 |
| R16_TIM1_AUX2 | 0x40012C54 | Auxiliary configuration register 2 | 0x0000 |
| R16_TIM1_CH5CVR | 0x40012C58 | Comparison register 5 | 0x0000 |
| R16_TIM1_CH6CVR | 0x40012C5C | Comparison register 6 | 0x0000 |
| R16_TIM1_CH7CVR | 0x40012C60 | Comparison register 7 | 0x0000 |
| R16_TIM1_CH8CVR | 0x40012C64 | Comparison register 8 | 0x0000 |

### 9.4.1 Control Register 1 (TIM1_CTLR1)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | CAPLVL | RW | In double-edge capture mode, the capture level indication is enabled.<br>1: Enable the indication function;<br>0: Disable the indication function.<br>*Note: When enabled, CHxCVR[16] indicates the level corresponding to the capture value.* | 0 |
| 14 | CAPOV | RW | Capture value mode configuration.<br>1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture;<br>0: The capture value is the value of the actual counter. | 0 |
| [13:10] | Reserved | RO | Reserved. | 0 |
| [9:8] | CKD[1:0] | RW | These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the dead time and the sampling clock used by the dead time generator and the | 0 |

| | | | | |
|---|---|---|---|---|
| | | | digital filter (ETR,TIx).<br>00: Tdts=Tck_int<br>01: Tdts = 2 × Tck_int<br>10: Tdts = 4 × Tck_int<br>11: Reserved. | |
| 7 | ARPE | RW | Auto-reload preload enable bit.<br>1: Enable Auto-reload Value Register (ATRLR).<br>0: Disable Auto-reload Value Register (ATRLR). | 0 |
| [6:5] | CMS[1:0] | RW | Center aligned mode selection.<br>00: Edge-aligned mode. The counter counts up or down based on the direction bit (DIR).<br>01: Center-aligned mode 1. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts down.<br>10: Center-aligned mode 2. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts up.<br>11: Center-aligned mode 3. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set when the counter counts both up and down.<br>*Note: When the counter is enabled (CEN=1), the transition from edge-aligned mode to center-aligned mode is not allowed.* | 0 |
| 4 | DIR | RW | Counting direction.<br>1: The counting mode of the counter is down counting;<br>0: The counting mode of the counter is up counting.<br>*Note: This bit is not valid when the counter is configured in center-aligned mode or encoder interface mode.* | 0 |
| 3 | OPM | RW | One-pulse mode.<br>1: The counter stops (Clear the CEN bit) when the next update event occurs.<br>0: The counter does not stop when the next update event occurs. | 0 |
| 2 | URS | RW | Update request source, by which the software selects the source of the UEV event.<br>1: If an update interrupt or DMA request is enabled, only an update interrupt or DMA request is generated if the counter overflows/underflows.<br>0: If an update interrupt or DMA request is enabled, an update interrupt or DMA request is generated by any of the following events.<br>-Counter overflow/underflow<br>-Setting the UG position<br>-Updates generated by the slave mode controller | 0 |
| 1 | UDIS | RW | Disable updates, the software allows/disables the generation of UEV events by means of this bit.<br>1: UEV is disabled. no update event is generated and the registers (ARR, PSC, CCRx) keep their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counters and prescaler are | 0 |

| | | | | |
|---|---|---|---|---|
| | | | reinitialized.<br>0: UEV is allowed. update (UEV) events are generated by any of the following events:<br>-Counter overflow/underflow<br>-Setting the UG position<br>- Updates generated by the slave mode controller<br>Registers with caches are loaded with their preloaded values. | |
| 0 | CEN | RW | Enables the counter.<br>1: Enable the counter.<br>0: Disable the counter.<br>*Note: The external clock, gated mode and encoder mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.* | 0 |

### 9.4.2 Control Register 2 (TIM1_CTLR2)

Offset address: 0x04

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | Reserved | RO | Reserved. | 0 |
| 14 | OIS4 | RW | Output idle state 4.<br>1: When MOE=0, if OC4N is implemented, OC1=1 after deadband;<br>0: When MOE=0, if OC4N is implemented, OC1=0 after deadband.<br>*Note: This bit cannot be modified after LOCK (TIM1_BDTR register) level 1, 2 or 3 has been set.* | 0 |
| 13 | OIS3N | RW | Output idle state 3.<br>1: OC1N = 1 after the dead zone when MOE = 0.<br>0: When MOE=0, OC1N=0 after dead zone.<br>*Note: This bit cannot be modified after the LOCK (TIM1_BDTR register) level 1, 2 or 3 has been set.* | 0 |
| 12 | OIS3 | RW | Output idle state 3, see OIS4. | 0 |
| 11 | OIS2N | RW | Output idle state 2, see OIS3N. | 0 |
| 10 | OIS2 | RW | Output idle state 2, see OIS4. | 0 |
| 9 | OIS1N | RW | Output idle state 1, see OIS3N. | 0 |
| 8 | OIS1 | RW | Output idle state 1, see OIS4. | 0 |
| 7 | TI1S | RW | TI1 selection.<br>1: TIM1_CH1, TIM1_CH2 and TIM1_CH3 pins connected to TI1 input (XOR combination).<br>0: TIM1_CH1 pin is connected directly to TI1 input. | 0 |
| [6:4] | MMS[2:0] | RW | Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in master mode. The possible combinations are as follows.<br>000: The UG bit of the Reset-TIM1_EGR register is used as the trigger output (TRGO). In the case of a reset generated by a trigger input (From a mode controller in reset mode), there is a delay in the signal on TRGO relative to the actual reset.<br>001: Enable - The counter enable signal CNT_EN is used as a trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or to control the enable from timers over a period of time. The counter enable signal is generated by the logical or | 0 |

| | | | | |
|---|---|---|---|---|
| | | | of the trigger input signal in CEN control bit and gated mode. When the counter enable signal is controlled by a trigger input, there is a delay on TRGO unless master/slave mode is selected (See the description of the MSM bit in the TIM1_SMCR register).<br>010: Update - The update event is selected as a trigger input (TRGO). For example, the clock of a master timer may be used as a prescaler for a slave timer.<br>011: comparison pulse - on the occurrence of a capture or a successful comparison, when the CC1IF flag is to be set (Even if it is already high), the trigger output sends a positive pulse (TRGO).<br>100: The comparison-OC1REF signal is used as a trigger output (TRGO).<br>101: The comparison-OC2REF signal is used as a trigger output (TRGO).<br>110: The comparison-OC3REF signal is used as a trigger output (TRGO).<br>111: The compare-OC4REF signal is used as the trigger output (TRGO). | |
| 3 | CCDS | RW | Capture compare DMA selection.<br>1: Send a DMA request for CHxCVR when an update event occurs.<br>0: Generate a DMA request for CHxCVR when CHxCVR occurs. | 0 |
| 2 | CCUS | RW | Compare capture control update selection bits.<br>1: If CCPC is set, they can be updated by setting the COM bit or a rising edge on TRGI.<br>0: If the CCPC is set, they can only be updated by setting the COM bit.<br>*Note: This bit only works for channels with complementary outputs.* | 0 |
| 1 | Reserved | RO | Reserved. | 0 |
| 0 | CCPC | RW | Compare capture preload control bits.<br>1: The CCxE, CCxNE and OCxM bits are preloaded and when this bit is set they are only updated when the COM bit is set.<br>0: CCxE, CCxNE and OCxM bits are not preloaded.<br>*Note: This bit only works for channels with complementary outputs.* | t |

### 9.4.3 Slave Mode Control Register (TIM1_SMCFGR)

Offset address: 0x08

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | ETP | RO | ETR trigger polarity selection, this bit selects whether to input ETR directly or to input the inverse of ETR.<br>1: Invert ETR, low or falling edge active;<br>0: ETR, active high or rising edge. | 0 |
| 14 | ECE | RW | External clock mode 2 enable selection.<br>1: Enables external clock mode 2.<br>0: Disable external clock mode 2.<br>*Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; however, TRGI cannot be connected to ETRF in this case (TS bit cannot be '111').*<br>*Note 2: When both external clock mode 1 and external* | 0 |

| | | | | |
|---|---|---|---|---|
| | | | *clock mode 2 are enabled, the external clock input is ETRF.* | |
| [13:12] | ETPS[1:0] | RW | The external trigger signal (ETRP) divides the frequency of this signal, which cannot exceed a maximum of 1/4 of the TIM1CLK frequency, and can be downconverted through this domain.<br>00: Prescaler off.<br>01: ETRP frequency divided by 2.<br>10: ETRP frequency divided by 4.<br>11: ETRP frequency divided by 8. | 0 |
| [11:8] | ETF[3:0] | RW | Externally triggered filtering, in fact, the digital filter is an event counter, which uses a certain sampling frequency to record up to N events and then produces a jump in the output.<br>0000: No filter, Fdts sampling;<br>0001: Sampling frequency Fsampling=Fck_int, N=2;<br>0010: Sampling frequency Fsampling=Fck_int, N=4;<br>0011: Sampling frequency Fsampling=Fck_int, N=8;<br>0100: Sampling frequency Fsampling = Fdts/2, N = 6;<br>0101: Sampling frequency Fsampling = Fdts/2, N = 8;<br>0110: Sampling frequency Fsampling = Fdts/4, N = 6;<br>0111: Sampling frequency Fsampling = Fdts/4, N = 8;<br>1000: Sampling frequency Fsampling = Fdts/8, N = 6;<br>1001: Sampling frequency Fsampling = Fdts/8, N = 8;<br>1010: Sampling frequency Fsampling = Fdts/16, N = 5;<br>1011: Sampling frequency Fsampling = Fdts/16, N = 6;<br>1100: Sampling frequency Fsampling = Fdts/16, N = 8;<br>1101: Sampling frequency Fsampling = Fdts/32, N = 5;<br>1110: Sampling frequency Fsampling = Fdts/32, N = 6;<br>1111: Sampling frequency Fsampling=Fdts/32, N=8. | 0 |
| 7 | MSM | RW | Master/slave mode selection.<br>1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is useful when the synchronization of several timers to a single external event is required.<br>0: Does not function. | 0 |
| [6:4] | TS[2:0] | RW | Trigger selection field, these 3 bits select the trigger input source used to synchronize the counter.<br>000: Internal trigger 0 (ITR0).<br>001: Internal trigger 1 (ITR1).<br>010: Internal trigger 2 (ITR2).<br>011: Internal trigger 3 (ITR3).<br>100: Edge detector of TI1 (TI1F_ED).<br>101: Filtered timer input 1 (TI1FP1).<br>110: Filtered timer input 2 (TI2FP2).<br>111: External trigger input (ETRF).<br>The above only changes when SMS is 0. | 0 |
| 3 | Reserved | RO | Reserved. | 0 |
| [2:0] | SMS[2:0] | RW | Input mode selection field. Selects the clock and trigger mode of the core counter.<br>000: Driven by the internal clock CK_INT.<br>001: Encoder mode 1, where the core counter increments or decrements the count at the edge of TI2FP2 depending on the level of TI1FP1.<br>010: Encoder mode 2, where the core counter increments or decrements the count at the edge of TI1FP1, depending on the level of TI2FP2. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 011: Encoder mode 3, where the core counter increments and decrements the count on the edges of TI1FP1 and TI2FP2 depending on the input level of another signal;<br>100: reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers.<br>101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled.<br>110: Trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled.<br>111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter. | |

### 9.4.4 DMA/Interrupt Enable Register (TIM1_DMAINTENR)

Offset address: 0x0C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | Reserved | RO | Reserved | 0 |
| 14 | TDE | RW | Trigger DMA request enable bit.<br>1: Enable trigger DMA request;<br>0: Disable trigger DMA request. | 0 |
| 13 | COMDE | RW | COM DMA request enable bit.<br>1: Allow COM DMA requests;<br>0: Disable COM DMA requests. | 0 |
| 12 | CC4DE | RW | Compare/capture channel 4 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 4;<br>0: Disable the DMA request of compare capture channel 4. | 0 |
| 11 | CC3DE | RW | Compare/capture channel 3 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 3;<br>0: Disable the DMA request of compare capture channel 3. | 0 |
| 10 | CC2DE | RW | Compare/capture channel 2 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 2;<br>0: Disable the DMA request of compare capture channel 2. | 0 |
| 9 | CC1DE | RW | Compare/capture channel 1 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 1;<br>0: Disable the DMA request of compare capture channel 1. | 0 |
| 8 | UDE | RW | Update DMA request enable bit.<br>1: Enable the updated DMA request<br>0: Disable the updated DMA request. | 0 |
| 7 | BIE | RW | Brake interrupt enable bit.<br>1: Enable the brake interrupt;<br>0: Disable the brake interrupt. | 0 |
| 6 | TIE | RW | Trigger interrupt enable bit<br>1: Enable the trigger interrupt;<br>0: Disable the trigger interrupt. | 0 |

| | | | | |
|---|---|---|---|---|
| 5 | COMIE | RW | COM interrupt enable bit<br>1: Enable COM interrupt;<br>0: Disable COM interrupt. | 0 |
| 4 | CC4IE | RW | Compare capture channel 4 interrupt enable bit<br>1: Enable compare capture channel 4 interrupt;<br>0: Disable compare capture channel 4 interrupt. | 0 |
| 3 | CC3IE | RW | Compare capture channel 3 interrupt enable bit<br>1: Enable compare capture channel 3 interrupt;<br>0: Disable compare capture channel 3 interrupt. | 0 |
| 2 | CC2IE | RW | Compare capture channel 2 interrupt enable bit<br>1: Enable compare capture channel 2 interrupt;<br>0: Disable compare capture channel 2 interrupt. | 0 |
| 1 | CC1IE | RW | Compare capture channel 1 interrupt enable bit<br>1: Enable compare capture channel 1 interrupt;<br>0: Disable compare capture channel 1 interrupt. | 0 |
| 0 | UIE | RW | Update interrupt enable bit<br>1: Enable updated interrupt;<br>0: Disable updated interrupt. | 0 |

### 9.4.5 Interrupt Status Register (TIM1_INTFR)

Offset address: 0x10

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:13] | Reserved | RO | Reserved | 0 |
| 12 | CC4OF | RW0 | Compare capture channel 4 over-capture flag bit. | 0 |
| 11 | CC3OF | RW0 | Compare capture channel 3 over-capture flag bit. | 0 |
| 10 | CC2OF | RW0 | Compare capture channel 2 over-capture flag bit. | 0 |
| 9 | CC1OF | RW0 | Compare capture channel 1 over-capture flag, only used when the compare capture channel is configured in input capture mode. This flag is set by hardware and cleared by software by writing 0.<br>1: When the counter value is captured into the capture compare register, the status of CC1IF has been set;<br>0: No over-capture occurs. | 0 |
| 8 | Reserved | RO | Reserved | 0 |
| 7 | BIF | RW0 | The brake interrupt flag bit, once the brake input is valid, by hardware for this position bit, can be cleared by software.<br>1: A set valid level is detected on the brake pin input.<br>0: No braking event is generated. | 0 |
| 6 | TIF | RW0 | Trigger interrupt flag bit, when a trigger event occurs by hardware to this location bit, by software to clear. Trigger events include the detection of a valid edge at the TRGI input from a mode other than gated, or any edge in gated mode.<br>1: Trigger event generation.<br>0: No trigger event is generated. | 0 |
| 5 | COMIF | RW0 | COM interrupt flag bit, this bit is set by hardware and cleared by software once a COM event is generated. com events including CCxE, CCxNE, OCxM are updated.<br>1: COM event generation.<br>0: No COM event is generated. | 0 |
| 4 | CC4IF | RW0 | Compare capture channel 4 interrupt flag bit. | 0 |
| 3 | CC3IF | RW0 | Compare capture channel 3 interrupt flag bit. | 0 |
| 2 | CC2IF | RW0 | Compare capture channel 2 interrupt flag bit. | 0 |

| | | | | |
|---|---|---|---|---|
| 1 | CC1IF | RW0 | Compare capture channel 1 interrupt flag bit.<br>If the compare capture channel is configured in output mode.<br>This bit is set by hardware when the counter value matches the comparison value, except in centrosymmetric mode. This bit is cleared by software.<br>1: The value of the core counter matches the value of compare capture register 1;<br>0: No match occurs.<br>If compare capture channel 1 is configured as input mode. This bit is set by hardware when a capture event occurs, and it is cleared by software or by reading the compare capture register.<br>1: The counter value has been captured compare capture register 1.<br>0: No input capture is generated. | 0 |
| 0 | UIF | RW0 | Update interrupt flag bit, this bit is set by hardware when an update event is generated and cleared by software.<br>1: Update interrupt generation.<br>0: No update event is generated.<br>The following scenarios generate update events.<br>If UDIS = 0, when the repeat counter value overflows or underflows.<br>If URS = 0, UDIS = 0, when the UG bit is set, or when the counter core counter is reinitialized by software.<br>If URS = 0, UDIS = 0, when the counter CNT is reinitialized by a trigger event. | 0 |

### 9.4.6 Event Generation Register (TIM1_SWEVGR)

Offset address: 0x14

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | BG | TG | COMG | CC4G | CC3G | CC2G | CC1G | UG |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved. | 0 |
| 7 | BG | WO | The brake event generation bit, which is set and cleared by software, is used to generate a brake event.<br>1: Generate a brake event. At this point, MOE=0, BIF=1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA are generated.<br>0: No action. | 0 |
| 6 | TG | WO | The trigger event generation bit, which is set by software and cleared by hardware, is used to generate a trigger event.<br>1: Generate a trigger event, TIF is set, and the corresponding interrupts and DMAs are generated if enabled.<br>0: No action. | 0 |
| 5 | COMG | WO | Compare capture control update generation bit. Generates a compare capture control update event. This bit is set by software and automatically cleared by hardware. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: When CCPC = 1, allow updating of CCxE, CCxNE, OCxM bits.<br>0: No action.<br>*Note: This bit is only valid for channels with complementary outputs (Channels 1, 2, 3).* | |
| 4 | CC4G | WO | Compare capture event generation bit 4. generates compare capture event 4. | 0 |
| 3 | CC3G | WO | Compare capture event generation bit 3. generates compare capture event 3. | 0 |
| 2 | CC2G | WO | Compare capture event generation bit 2. generates compare capture event 2. | 0 |
| 1 | CC1G | WO | Compare capture event generation bit 1. generates compare capture event 1.<br>This bit is set by software and cleared by hardware. It is used to generate a compare capture event.<br>1: Generate a compare capture event on compare capture channel 1.<br>If compare capture channel 1 is configured as output.<br>Set the CC1IF bit. Generate the corresponding interrupts and DMAs if they are enabled.<br>If compare capture channel 1 is configured as input.<br>The current core counter value is captured to compare capture register 1; set the CC1IF bit to generate the corresponding interrupts and DMAs if they are enabled; if CC1IF is already set, set the CC1OF bit.<br>0: No action. | 0 |
| 0 | UG | WO | Update event generation bit to generate an update event. This bit is set by software and is automatically cleared by hardware.<br>1: Initialize the counter and generate an update event.<br>0: No action.<br>*Note: The prescaler counter is also cleared to zero, but the prescaler factor remains unchanged. The core counter is cleared if in centrosymmetric mode or incremental counting mode; if in decremental counting mode, the core counter takes the value of the reload value register.* | 0 |

### 9.4.7 Compare/Capture Control Register 1 (TIM1_CHCTLR1)

Offset address: 0x18

The channel can be used in input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register have different roles in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC2CE | OC2M[2:0] | | | OC2PE | OC2FE | CC2S[1:0] | | OC1CE | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |
| IC2F[3:0] | | | | IC2PSC[1:0] | | | | IC1F[3:0] | | | | IC1PSC[1:0] | | | |

Compare mode (pin direction is output).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | OC2CE | RW | Compare capture channel 2 clear enable bit.<br>1: Clear OC2REF bit zero once ETRF input is detected | 0 |

| | | | | |
|---|---|---|---|---|
| | | | high;<br>0: OC2REF is not affected by ETRF input. | |
| [14:12] | OC2M[2:0] | RW | Compare Capture Channel 2 mode setting field.<br>The 3 bits define the action of the output reference signal OC2REF, which determines the values of OC2, OC2N. OC2REF is active high, while the active levels of OC2 and OC2N depend on the CC2P, CC2NP bits.<br>000: Freeze. Comparison of the value of the capture register with the value of the comparison between the core counters does not work for OC1REF.<br>001: Force to set to valid level. Forcing OC1REF high when the core counter has the same value as the comparison capture register 2.<br>010: Force to set to invalid level. Forcing OC1REF low when the value of the core counter is the same as the comparison capture register 2.<br>011: Flip. Flips the level of OC1REF when the core counter is the same as the value of compare capture register 2.<br>100: Forced to invalid level. Forces OC1REF to low.<br>101: Forced to valid level. Force OC1REF to high.<br>110: PWM mode 1: When counting up, once the core counter is less than the value of the comparison capture register, channel 2 is an effective level, otherwise it is an invalid level; When counting down, once the core counter is greater than the value of the comparison capture register, channel 2 is at an invalid level (OC2REF=0), otherwise it is at an effective level (OC2REF = 1);<br>111: PWM mode 2: When counting up, once the core counter is less than the value of the comparison capture register, channel 2 is at an invalid level, otherwise it is at an effective level; When counting down, once the core counter is greater than the value of the comparison capture register, Channel 2 is active (OC2REF=1), otherwise it is inactive (OC2REF=0).<br>*Note: (1) This bit cannot be modified once the LOCK level is set to 3 and CC1S=00b. In PWM mode 1 or PWM mode 2, the OC1REF level is changed only when the comparison result is changed or when switching from freeze mode to PWM mode in the output comparison mode.*<br>*(2) When using the high-frequency clock counting mode (enabling the HSCK_MODE bit), the timer only supports PWM mode 1(110b) or PWM mode 2(111b) of the OCxM bit.* | 0 |
| 11 | OC2PE | RW | Compare Capture Register 2 preload enable bit.<br>1: Enable the preload function of compare capture register 2, read and write operations only operate on the preload registers, the preload value of compare capture register 2 is loaded into the current shadow register when the update event comes;<br>0: Disable the preload function of compare capture register 2, compare capture register 2 can be written at any time, and the newly written value takes effect immediately.<br>*Note: Once the LOCK level is set to 3 and CC2S=00,* | 0 |

| | | | | |
|---|---|---|---|---|
| | | | *this bit cannot be modified; PWM mode can be used only in single pulse mode (OPM=1) without confirming the pre-load register, otherwise its action is not determined.* | |
| 10 | OC2FE | RW | Compare Capture Channel 2 fast enable bit, this bit is used to speed up the response of the compare capture channel output to a trigger input event.<br>1: The active edge of the input to the flipflop acts as if a comparison match has occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the valid edge of the sample trigger and the output of the compare capture channel 2 is reduced to 3 clock cycles.<br>0: Based on the value of the counter and compare capture register 1, compare capture channel 2 operates normally, even if the flip-flop is open. The minimum delay to activate the compare capture channel 2 output is 5 clock cycles when the input of the flipflop has a valid edge.<br>OC2FE only works when the channel is configured to PWM1 or PWM2 mode. | 0 |
| [9:8] | CC2S[1:0] | RW | Compare capture channel 2 input selection fields.<br>00: Compare capture channel 2 is configured as an output.<br>01: Compare capture channel 2 is configured as an input and IC2 is mapped on TI2.<br>10: Compare capture channel 2 is configured as an input and IC2 is mapped on TI1.<br>11: Compare Capture Channel 2 is configured as an input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit).<br>*Note: Compare Capture Channel 2 is writable only when the channel is off (When CC2E is zero).* | 0 |
| 7 | OC1CE | RW | Compare capture channel 1 clear enable bit. | 0 |
| [6:4] | OC1M[2:0] | RW | Compare capture channel 1 mode setting field. | 0 |
| 3 | OC1PE | RW | Compare capture register 1 preload enable bit. | 0 |
| 2 | OC1FE | RW | Compare capture channel 1 fast enable bit. | 0 |
| [1:0] | CC1S[2:0] | RW | Compare capture channel 1 input selection fields. | 0 |

Capture mode (pin direction is input).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | IC2F[3:0] | RW | The input capture filter 2 configuration field, these bits set the sampling frequency of the TI1 input and the digital filter length. The digital filter consists of an event counter, which records N events and then generates a jump in the output.<br>0000: No filter, sampled at fDTS.<br>1000: Sampling frequency Fsampling = Fdts/8, N = 6.<br>0001: Sampling frequency Fsampling=Fck_int, N=2.<br>1001: Sampling frequency Fsampling = Fdts/8, N = 8.<br>0010: Sampling frequency Fsampling=Fck_int, N=4.<br>1010: Sampling frequency Fsampling = Fdts/16, N = 5.<br>0011: Sampling frequency Fsampling=f=Fck_int, N=8.<br>1011: Sampling frequency Fsampling = Fdts/16, N = 6. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0100: Sampling frequency Fsampling = Fdts/2, N = 6.<br>1100: Sampling frequency Fsampling = Fdts/16, N = 8.<br>0101: Sampling frequency Fsampling = Fdts/2, N = 8.<br>1101: Sampling frequency Fsampling = Fdts/32, N = 5.<br>0110: Sampling frequency Fsampling = Fdts/4, N = 6.<br>1110: Sampling frequency Fsampling = Fdts/32, N = 6.<br>0111: Sampling frequency Fsampling = Fdts/4, N = 8.<br>1111: Sampling frequency Fsampling=Fdts/32, N=8. | |
| [11:10] | IC2PSC[1:0] | RW | Compare capture channel 2 prescaler configuration field, these 2 bits define the prescaler coefficient for compare capture channel 2. Once CC1E = 0, the prescaler is reset.<br>00: Without prescaler, one capture is triggered for each edge detected on the capture input.<br>01: Capture triggered every 2 events.<br>10: Capture triggered every 4 events.<br>11: Capture is triggered every 8 events. | 0 |
| [9:8] | CC2S[1:0] | RW | Compare the capture channel 2 input selection field, these 2 bits define the direction of the channel (Input/output), and the selection of the input pin.<br>00: Compare capture channel 1 channel is configured as an output.<br>01: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TI1.<br>10: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TI2.<br>11: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (By the TS bit).<br>*Note: CC1S is writable only when the channel is off (CC1E is 0).* | 0 |
| [7:4] | IC1F[3:0] | RW | Input capture filter 1 configuration field. | 0 |
| [3:2] | IC1PSC[1:0] | RW | Compare capture channel 1 prescaler configuration field. | 0 |
| [1:0] | CC1S[1:0] | RW | Compare capture channel 1 input selection field. | 0 |

### 9.4.8 Compare/Capture Control Register 2 (TIM1_CHCTLR2)

Offset address: 0x1C

The channel can be used in input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register serve different purposes in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC4CE | OC4M[2:0] | | | OC4PE | OC4FE | CC4S[1:0] | | OC3CE | OC3M[2:0] | | | OC3PE | OC3FE | CC3S[1:0] | |
| IC4F[3:0] | | | | IC4PSC[1:0] | | | | IC3F[3:0] | | | | IC3PSC[1:0] | | | |

Compare mode (Pin direction is output).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | OC4CE | RW | Compare capture channel 4 clear enable bit. | 0 |
| [14:12] | OC4M[2:0] | RW | Compare capture channel 4 mode setting field. | 0 |
| 11 | OC4PE | RW | Compare capture register 4 preload enable bit. | 0 |

| | | | | |
|---|---|---|---|---|
| 10 | OC4FE | RW | Compare capture channel 4 fast enable bit. | 0 |
| [9:8] | CC4S[1:0] | RW | Compare capture channel 4 input selection fields. | 0 |
| 7 | OC3CE | RW | Compare capture channel 3 clear enable bit. | 0 |
| [6:4] | OC3M[2:0] | RW | Compare capture channel 3 mode setting field. | 0 |
| 3 | OC3PE | RW | Compare capture register 3 preload enable bit. | 0 |
| 2 | OC3FE | RW | Compare capture channel 3 fast enable bit. | 0 |
| [1:0] | CC3S[1:0] | RW | Compare capture channel 3 input selection fields. | 0 |

Capture mode (pin direction is input).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | IC4F[3:0] | RW | Input capture filter 4 configuration field. | 0 |
| [11:10] | IC4PSC[1:0] | RW | Compare capture channel 4 prescaler configuration field. | 0 |
| [9:8] | CC4S[1:0] | RW | Compare capture channel 4 input selection fields. | 0 |
| [7:4] | IC3F[3:0] | RW | Input capture filter 3 configuration field. | 0 |
| [3:2] | IC3PSC[1:0] | RW | Compare capture channel 3 prescaler configuration fields. | 0 |
| [1:0] | CC3S[1:0] | RW | Compare capture channel 3 input selection fields. | 0 |

### 9.4.9 Compare/Capture Enable Register (TIM1_CCER)

Offset address: 0x20

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:14] | Reserved | RO | Reserved | 0 |
| 13 | CC4P | RW | Compare capture channel 4 output polarity setting bit. | 0 |
| 12 | CC4E | RW | Compare capture channel 4 output enable bit. | 0 |
| 11 | CC3NP | RW | Compare capture channel 3 complementary output polarity setting bit. | 0 |
| 10 | CC3NE | RW | Compare capture channel 3 complementary output enable bits. | 0 |
| 9 | CC3P | RW | Compare capture channel 3 output polarity setting bit. | 0 |
| 8 | CC3E | RW | Compare capture channel 3 output enable bit. | 0 |
| 7 | CC2NP | RW | Compare capture channel 2 complementary output polarity setting bit. | 0 |
| 6 | CC2NE | RW | Compare capture channel 2 complementary output enable bits. | 0 |
| 5 | CC2P | RW | Compare capture channel 2 output polarity setting bit. | 0 |
| 4 | CC2E | RW | Compare capture channel 2 output enable bit. | 0 |
| 3 | CC1NP | RW | Compare capture channel 1 complementary output polarity setting bit. | 0 |
| 2 | CC1NE | RW | Compare capture channel 1 complementary output enable bit. | 0 |
| 1 | CC1P | RW | Compare capture channel 1 output polarity setting bit.<br>CC1 channel is configured as output:<br>1: OC1 is active low;<br>0: OC1 is active high.<br>CC1 channel is configured as input:<br>This bit selects whether IC1 or the inverted signal of IC1 is used as the trigger or capture signal.<br>1: Inversion: Capture occurs at the falling edge of IC1; When used as an external trigger, IC1 is inverted;<br>0: Non-inverting: capture occurs at the rising edge of IC1; When used as an external trigger, IC1 does not invert.<br>*Note: Once the LOCK level (the LOCK bit in the TIM1_BDTR register) is set to 3 or 2, this bit cannot be* | 0 |

| | | | | |
|---|---|---|---|---|
| | | | *modified.* | |
| 0 | CC1E | RW | Compare capture channel 1 output enable bit.<br>CC1 channel is configured as output:<br>1: On. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE.<br>0: Off. OC1 prohibits output, so the output level of OC1 depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits.<br>CC1 channel is configured as input:<br>This bit determines whether the value of the counter can be captured in the TIM1_CH1CVR register.<br>1: Capture enabled;<br>0: Capture disabled. | 0 |

### 9.4.10 Counter for Advanced-control Timer (TIM1_CNT)

Offset address: 0x24

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CNT[15:0] | RW | The real-time value of the timer's counter. | 0 |

### 9.4.11 Counting Clock Prescaler (TIM1_PSC)

Offset address: 0x28

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | PSC[15:0] | RW | The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the divider/(PSC+1). | 0 |

### 9.4.12 Auto-reload Value Register (TIM1_ATRLR)

Offset address: 0x2C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | ARR[15:0] | RW | The value of this field will be loaded into the counter, see section 9.2.3 for when the ATRLR acts and updates; the counter stops when the ATRLR is empty. | 0xFFFF |

### 9.4.13 Repeat Count Value Register (TIM_RPTCR)

Offset address: 0x30

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved | 0 |
| [7:0] | REP[7:0] | RW | The value of the repeat counter. | 0 |

### 9.4.14 Compare/Capture Register 1 (TIM1_CH1CVR)

Offset address: 0x34

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH1CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL1 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH1CVR[15:0] | RW | Compare/capture the value of register channel 1. | 0 |

### 9.4.15 Compare/Capture Register 2 (TIM1_CH2CVR)

Offset address: 0x38

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL2 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH2CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL2 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH2CVR[15:0] | RW | Compare/capture the value of register channel 2. | 0 |

### 9.4.16 Compare/Capture Register 3 (TIM1_CH3CVR)

Offset address: 0x3C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL3 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH3CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL3 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH3CVR[15:0] | RW | Compare/capture the value of register channel 3. | 0 |

### 9.4.17 Compare/Capture Register 4 (TIM1_CH4CVR)

Offset address: 0x40

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH4CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL4 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH4CVR[15:0] | RW | Compare/capture the value of register channel 4. | 0 |

### 9.4.18 Brake and Dead-time Register (TIM1_BDTR)

Offset address: 0x44

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | MOE | RW | Main output enable bit. Once the brake signal is active, it will be cleared asynchronously.<br>1: Enable OCx and OCxN to be set as outputs.<br>0: Disable the output of OCx and OCxN or force to idle state. | 0 |
| 14 | AOE | RW | Auto output enable.<br>1: MOE can be set by software or set in the next update event.<br>0: MOE can only be set by software. | 0 |
| 13 | BKP | RW | The brake input polarity setting bit.<br>1: Brake input active high.<br>0: Brake input active low.<br>*Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bit requires an HB clock before it can take effect.* | 0 |
| 12 | BKE | RW | Brake function enable bit.<br>1: Brake input enabled.<br>0: Brake input disabled.<br>*Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bit requires an HB clock before it can take effect.* | 0 |
| 11 | OSSR | RW | 1: When the timer is not operating, once CCxE=1 or CCxNE=1, first turn on OC/OCN and outputinvalid level, then set OCx, OCxN enable output signal=1.<br>0: When the timer is not operating, OC/OCN output is disabled.<br>*Note: When LOCK level 1 is set, this bit cannot be modified.* | 0 |
| 10 | OSSI | RW | 1: When the timer is not operating, once CCxE = 1 or CCxNE = 1, OC/OCN first outputs its idle level, then OCx, OCxN enable output signal = 1.<br>0: When the timer is not operating, OC/OCN output is disabled.<br>*Note: When LOCK level 1 is set, this bit cannot be modified.* | 0 |
| [9:8] | LOCK[1:0] | RW | Lock the function setting field.<br>00: Disable the locking function.<br>01: Lock level 1, no DTG, BKE, BKP, AOE, OISx and OISxN bits can be written.<br>10: Lock level 2, where the bits in lock level 1 cannot be written, nor the CC polarity bits, nor the OSSR and OSSI bits.<br>11: Lock level 3, cannot write to the bits in lock level 2, and cannot write to the CC control bits.<br>*Note: After system reset, the LOCK bit can only be written once and cannot be modified again until reset.* | 0 |

| | | | | |
|---|---|---|---|---|
| [7:0] | DTG[7:0] | RW | Dead-time setting bits that define the duration of the dead-time between complementary outputs.<br>Assume that DT denotes its duration.<br>DTG[7:5]=0xx=>DT=DTG[7:0]*Tdtg, Tdtg=TDTS;<br>DTG[7:5]=10x=>DT=(64+DTG[5:0])*Tdtg, Tdtg=2*TDTS;<br>DTG[7:5]=110=>DT=(32+DTG[4:0])*Tdtg, Tdtg =8×TDTS;<br>DTG[7:5]=111=>DT=(32+DTG[4:0])*Tdtg, Tdtg =16*TDTS. | 0 |

### 9.4.19 DMA Control Register (TIM1_DMACFGR)

Offset address: 0x48

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:13] | Reserved | RO | Reserved | 0 |
| [12:8] | DBL[4:0] | RW | The length of the DMA continuous transmission, the actual value of which is the value of this field + 1. | 0 |
| [7:5] | Reserved | RO | Reserved | 0 |
| [4:0] | DBA[4:0] | RW | These bits define the offset of the DMA in continuous mode from the address where control register 1 is located. | 0 |

### 9.4.20 DMA Address Register for Continuous Mode (TIM1_DMAADR)

Offset address: 0x4C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | DMAB [15:0] | RW | The address of the DMA in continuous mode. | 0 |

### 9.4.21 Auxiliary configuration Register 1 (TIM1_AUX1)

Offset address: 0x50

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | Reserved | RO | Reserved | 0 |
| 14 | HSCK_MODE | RW | Counter clock selection bit, which needs to turn on PLL clock to use this function:<br>1: The core counters of channels 1, 2 and 3 use PLL clock as working clock;<br>0: Channels 1, 2 and 3 maintain the original clock signal as the working clock. | 0 |
| 13 | CC3N_SPEC_EN | RW | The complementary channel of channel 3 outputs independent PWM function in edge alignment mode. Enable:<br>1: Complementary channel 3 output independent PWM is enabled, and the comparison value is set by CH8CVR register;<br>0: The function is off. | 0 |
| 12 | CC2N_SPEC_EN | RW | The complementary channel of channel 2 outputs independent PWM function in edge alignment mode. Enable:<br>1: Complementary channel 2 output independent PWM is enabled, and the comparison value is set by CH7CVR register; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0: The function is off. | |
| 11 | CC1N_SPEC_EN | RW | The complementary channel of channel 1 outputs independent PWM function in edge alignment mode. Enable:<br>1: Complementary channel 1 output independent PWM is enabled, and the comparison value is set by CH6CVR register;<br>0: The function is off. | 0 |
| 10 | ADC_CAP | RW | The function of capturing the signal edge of ADC module is enabled:<br>1. Alternate channel 1 realizes the function of capturing the double edge, and the counter is cleared after capturing;<br>0: The function is off, and the double-edge capture function of Channel 1 is normal. | 0 |
| 9 | CC5DE | RW | DMA request enable bits of compare channel 5:<br>1: DMA request of compare channel 5 is allowed;<br>0: DMA request of compare channel 5 is prohibited. | 0 |
| 8 | OC8PE | RW | Comparison register 8 preloads the enable bit. | 0 |
| 7 | OC7PE | RW | Comparison register 7 preloads the enable bit. | 0 |
| 6 | OC6PE | RW | Comparison register 6 preloads the enable bit. | 0 |
| 5 | OC5PE | RW | Comparison register 5 preloads the enable bit. | 0 |
| 4 | Reserved | RO | Reserved | 0 |
| 3 | CMS_SPEC | RW | Move function in center symmetric mode:<br>1: The central symmetric mode uses different comparison registers for the rising and falling stages (channel 1 rising stage CH1CVR, fall stage CH6CVR, channel 2, 3 and so on);<br>0: The function is turned off. | 0 |
| 2 | CAP_ED_CH4 | RW | Channel 4 bilateral capture function enables:<br>1: Enables channel 4's bilateral capture function;<br>0: Disable channel 4's bilateral capture function. | 0 |
| 1 | CAP_ED_CH3 | RW | Channel 3 bilateral capture function enables:<br>1: Enables channel 3's bilateral capture function;<br>0: Disable channel 3's bilateral capture function. | 0 |
| 0 | CAP_ED_CH2 | RW | Channel 2 bilateral capture function enables:<br>1: Enables channel 2's bilateral capture function;<br>0: Disable channel 2's bilateral capture function. | 0 |

### 9.4.22 Auxiliary configuration Register 2 (TIM1_AUX2)

Offset address: 0x54

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | CC5P | RW | Channel 5 outputs polarity configuration bits. | 0 |
| 14 | OC3NM | RW | Channel 3 complementary channel output independent PWM mode setting bit | 0 |
| 13 | OC2NM | RW | Channel 2 complementary channel output independent PWM mode setting bit | 0 |
| 12 | OC1NM | RW | Channel 1 complementary channel output independent PWM mode setting bit:<br>1: PWM mode 2: When counting upwards, once the core counter is less than the value of the comparison capture register, channel 1 is an invalid level, otherwise it is an active level; when counting downwards, once the core counter is greater than the value of the comparison capture register, channel 1 is an active level | 0 |

| | | | | |
|---|---|---|---|---|
| | | | (OC1REF=1), otherwise it is an invalid level (OC1REF=0);<br>0: PWM mode 1: When counting upwards, channel 1 is the valid level once the core counter is less than the value of the comparison capture register, otherwise it is an invalid level; when counting downwards, channel 1 is the invalid level (OC1REF=0), otherwise it is an active level (OC1REF=1). | |
| 11 | OC5CE | RW | Comparison channel 5 clear enable bit. | 0 |
| 10 | OC5M | RW | Compare channel 5 mode settings：<br>1: PWM mode 2: When counting upwards, once the core counter is less than the value of the comparison capture register, channel 5 is an invalid level, otherwise it is an active level; when counting downwards, once the core counter is greater than the value of the comparison capture register, channel 5 is an active level (OC5REF=1), otherwise it is an invalid level (OC5REF=0);<br>0: PWM mode 1: When counting upwards, channel 5 is the valid level once the core counter is less than the value of the comparison capture register, otherwise it is an invalid level; when counting downwards, channel 5 is the invalid level once the core counter is greater than the value of the comparison capture register, otherwise it is an active level (OC5REF=0), otherwise it is an active level (OC5REF=1). | 0 |
| 9 | DTN_MODE | RW | The dead time set by enabling DT_VLU2 occurs at the falling edge of OCXREF.<br>1: The dead time duration set by DT_VLU2 occurs on the falling edge of OCXREF; 0: The function is turned off, and the dead time duration of the falling edge of OCXREF is the original DTG set time. | 0 |
| 8 | DTP_MODE | RW | The dead time set by enabling DT_VLU2 occurs on the rising edge of OCXREF.<br>1: The dead time duration set by DT_VLU2 occurs at the rising edge of OCXREF;<br>0: The function is turned off and the dead time duration of the rising edge of OCXREF is the original DTG setting time. | 0 |
| [7:0] | DT_VLU2[7:0] | RW | When the dead time asymmetry function is realized, the setting time is less than the dead time duration of DTG, which should be used in conjunction with DTP_MODE and DTN_MODE bits.<br>*Note: The set dead time duration is the set value multiplied by the Tdtg set in DTG, which should be greater than 0 and less than the dead time set by DTG.* | 0 |

### 9.4.23 Comparison Register 5 (TIM1_CH5CVR)

Offset address: 0x58

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CH5CVR[15:0] | RW | The value of comparison register 5. | 0 |

### 9.4.24 Comparison Register 6(TIM1_CH6CVR)

Offset address: 0x5C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CH6CVR[15:0] | RW | The value of comparison register 6.<br>As the complementary channel 1 outputs the comparison value of the independent PWM function or the comparison value of the down phase of the channel 1 when the center symmetric mode is shifted toward the function. | 0 |

### 9.4.25 Comparison Register 7(TIM1_CH7CVR)

Offset address: 0x5C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CH6CVR[15:0] | RW | The value of comparison register 7.<br>As the complementary channel 2 outputs the comparison value of the independent PWM function or the comparison value of the down phase of the channel 2 when the center symmetric mode is shifted toward the function. | 0 |

### 9.4.26 Comparison Register 8(TIM1_CH8CVR)

Offset address: 0x5C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CH6CVR[15:0] | RW | The value of comparison register 8.<br>As the complementary channel 3 outputs the comparison value of the independent PWM function or the comparison value of the down phase of the channel 3 when the center symmetric mode is shifted toward the function. | 0 |

# Chapter 10 General-Purpose Timer (GPTM)

The general-purpose timer module contains a 16-bit auto-reloadable timer (TIM2), for measuring pulse width or generating pulses of a specific frequency, PWM waves, etc. It can be used in automation control, power supply, etc.

## 10.1 Main Features

The main features of the general-purpose timer include:

- 16-bit auto-reload counter, supports incremental counting mode
- 16-bit prescaler with dynamically adjustable crossover factor from 1 to 65536
- Support 4 independent comparison capture channels
- Each compare capture channel supports multiple operating modes, such as: input capture, output compare, PWM generation, and single pulse output
- Support simple deadtime generation, complementary outputs by multiplexing the remaining channel outputs
- Support external signal control timer
- Support DMA in multiple modes.
- Support incremental code, cascade and synchronization between timers.

## 10.2 Principle and Structure

Figure 10-1 Block diagram of the structure of the general-purpose timer

### 10.2.1 Overview

As shown in Figure 10-1, the structure of the general-purpose timer can be roughly divided into three parts, namely the input clock part, the core counter part and the compare capture channel part.

The clock for the general-purpose timer can come from the HB bus clock (CK_INT), from the external clock input pin (TIMx_ETR), from other timers with clock output (ITRx), and from the input of the compare capture channel (TIMx_CHx). These input clock signals become CK_PSC clocks after various set filtering and dividing operations, etc., and are output to the core counter section. In addition, these complex clock sources can also be output as TRGO

to other peripherals such as timers and ADCs.

The core of the general-purpose timer is a 16-bit counter (CNT). CK_PSC is divided by a prescaler (PSC) to become CK_CNT and then finally fed to the CNT, which supports incremental counting mode, decremental counting mode, and incremental and decremental counting mode, and has an auto-reload register (ATRLR) to reload the initialization value for the CNT at the end of each counting cycle.

The general-purpose timer has four sets of compare capture channels, each of which can input pulses from exclusive pins or output waveforms to pins, i.e., the compare capture channels support both input and output modes. The input of each channel of the compare capture register supports filtering, dividing, edge detection, and other operations, and supports mutual triggering between channels, and can also provide clock for the core counter CNT. Each comparison capture channel has a set of comparison capture registers (CHxCVR) that support comparison with the main counter (CNT) to output pulses.

### 10.2.2 Differences between General-Purpose Timer and Advanced-Control Timer

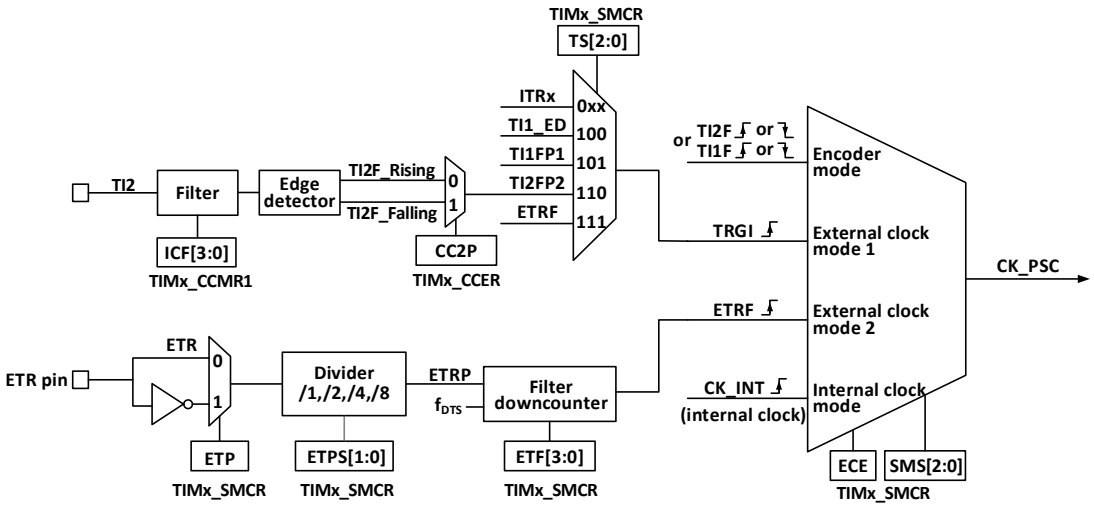
Compared to advanced-control timers, general-purpose timers lack the following features.

1) The general-purpose timer lacks a repeat count register for counting the count cycles of the core counter.
2) The comparison capture channel of the general-purpose timer lacks deadband generation and has no complementary output.
3) The general-purpose timer does not have a brake signal mechanism.

### 10.2.3 Clock Input

This section discusses the source of CK_PSC. The clock source portion of the overall block diagram of the general-purpose timer is captured here.

Figure 10-2 General-Purpose timer CK_PSC source block diagram




The optional input clocks can be divided into 4 categories:

1) Route of the external clock pin (ETR) input: ETR → ETRP → ETRF.
2) Internal HB clock input route: CK_INT.

3) Route from the compare capture channel pin (TIM2_CHx): TIM2_CHx → TIx → TIxFPx.
4) Input from other internal timers: ITRx.

The actual operation can be divided into 3 categories by determining the choice of input pulse for the SMS of the CK_PSC source.
1) Selection of the internal clock source (CK_INT).
2) External clock source mode 1.
3) External clock source mode 2.
All 4 clock source sources mentioned above can be selected by these 4 operations.

#### 10.2.3.1 Internal Clock Source (CK_INT)
If the general-purpose timer is started when the SMS field is held at 000b, then it is the internal clock source (CK_INT) that is selected as the clock. At this point CK_INT is CK_PSC.

#### 10.2.3.2 External Clock Source Mode 1
When the SMS domain is set to 111b, external clock source mode 1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source for CK_PSC. it is worth noting that the user also needs to select the source for TRGI by configuring the TS domain. the TS domain can select the following types of pulses as clock sources:
1) Internal trigger (ITRx, x is 0,1,2,3).
2) Comparison of the signal after capturing channel 1 through the edge detector (TI1F_ED).
3) Comparison of signals TI1FP1, TI2FP2 of the capture channel.
4) The signal ETRF from the external clock pin input.

#### 10.2.3.3 External Clock Source Mode 2
Using external trigger mode 2 can count every rising edge or falling edge of the external clock pin input. When the ECE bit is set, external clock source mode 2 is used. When using external clock source mode 2, ETRF is selected as CK_PSC. The ETR pin becomes ETRP after passing through an optional inverter (ETP) and frequency divider (ETPS), and then becomes ETRF after passing through a filter (ETF).

When ECE is set and SMS is set to 111b, it is equivalent to TS selecting ETRF as input.

#### 10.2.3.4 Encoder Mode
Setting SMS to 001b, 010b, 011b will enable encoder mode. When the encoder mode is enabled, you can choose to output the signal at a certain level of TI1FP1 and TI2FP2 with the other transition edge as the signal. This mode is used when an external encoder is used. Refer to Section 10.3.7 for specific functions.

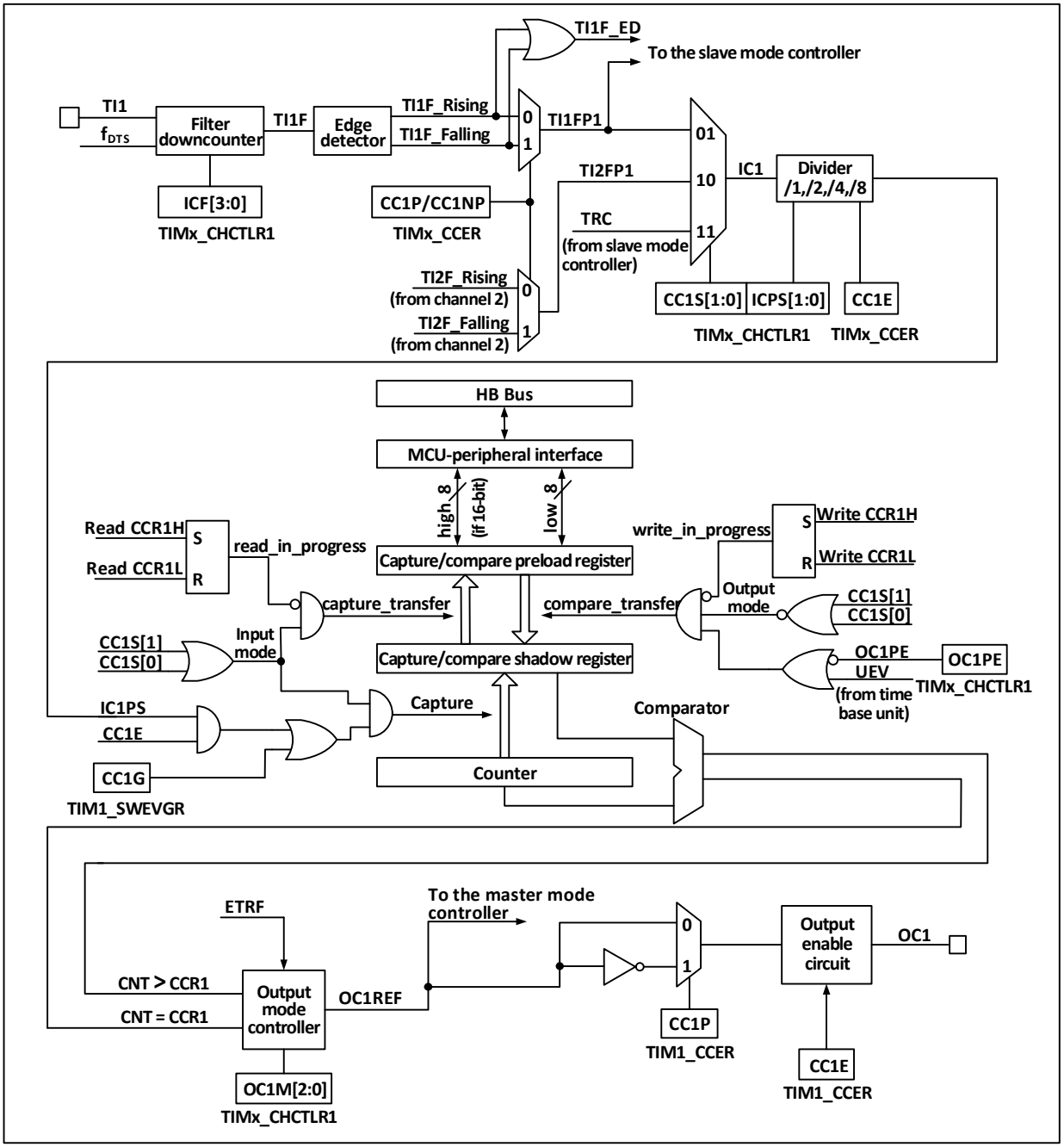
### 10.2.4 Counters and Peripherals
CK_PSC is input to the prescaler (PSC) for dividing. the PSC is 16-bit and the actual dividing factor is equal to the value of R16_TIMx_PSC + 1. CK_PSC goes through the PSC and becomes CK_INT. changing the value of R16_TIM1_PSC does not take effect in real time, but is updated to the PSC after an update event. the update event includes a UG bit clear and reset.

The core of the timer is a 16-bit counter (CNT). CK_CNT will eventually be input to CNT. CNT supports count-up mode, count-down mode and count-up/down mode, and there is an automatic reset value register (ATRLR) to reload the initial value for CNT after each counting period.

### 10.2.5 Compare/capture Channels

The core of the compare capture channel, which is the core of the timer to achieve complex functions, is the compare capture register, supplemented by digital filtering, frequency division and inter-channel multiplexing in the peripheral input section, and comparator and output control in the output section. The structure block diagram of the compare capture channel is shown in Figure 10-3.

Figure 10-3 Block diagram of the structure of the compare capture channel




The signal is input from the channel x pin and optionally made as TIx (The source of TI1 can be more than CH1, see block diagram 10-1 of the timer), TI1 is passed through the filter (ICF[3:0]) to generate TI1F, and then divided into TI1F_Rising and TI1F_Falling through the edge detector, these two signals are selected (CC1P) to generate TI1FP1, TI1FP1 and TI2FP1 from channel 2 are sent together to CC1S to select to become IC1, which is sent to the

comparison capture register after ICPS dividing.

The compare capture register consists of a preload register and a shadow register, and the read/write process operates only on the preload register. In capture mode, the capture occurs on the shadow register and is then copied to the preload register; in compare mode, the contents of the preload register are copied to the shadow register, and then the contents of the shadow register are compared to the core counter (CNT).

## 10.3 Function and Implementation

The complex functions of a general-purpose timer are implemented by manipulating the timer's compare capture channel, clock input circuitry, and counter and peripheral components. The clock input to the timer can be derived from multiple clock sources including the input to the compare capture channel. The operation of the compare capture host channel and clock source selection directly determines its function. The compare capture channel is bidirectional and can operate in both input and output modes.

### 10.3.1 Input Capture Mode

The input capture mode is one of the basic functions of the timer. The principle of input capture mode is that when a determined edge on the ICxPS signal is detected, a capture event is generated and the current value of the counter is latched into the compare capture register (R16_TIM2_CHCTLRx). The CCxIF (In R16_TIM2_INTFR) is set when a capture event occurs, and the corresponding interrupt or DMA is generated if enabled. If the CCxIF is already set when a capture event occurs, the CCxOF bit is set. the CCxIF can be cleared by software, or by hardware by reading the compare capture register. CCxOF is cleared by software.

An example of channel 1 to illustrate the steps to use the input capture mode is as follows:

1) Configure the CCxS domain to select the source of the ICx signal. For example, set it to 10b and select TI1FP1 as the source of IC1, not using the default setting, the CCxS domain defaults to making the comparison capture module the output channel.
2) Configure the ICxF domain to set the digital filter for the TI signal. The digital filter will sample the signal at a determined frequency, a determined number of times, and then output a hop. This sampling frequency and number of times is determined by ICxF.
3) Configure the CCxP bit to set the polarity of the TIxFPx. For example, keeping the CC1P bit low and selecting rising edge jumps.
4) Configure the ICxPS domain to set the ICx signal to be the crossover factor between ICxPS. For example, keeping ICxPS at 00b, without crossover.
5) Configure the CCxE bit to allow capturing the value of the core counter (CNT) into the compare capture register. Set the CC1E bit.
6) Configure the CCxIE and CCxDE bits as needed to determine whether to allow enable interrupts or DMA.

This completes the comparison capture channel configuration.

When a captured pulse is input to TI1, the value of the core counter (CNT) is recorded in the compare capture register, CC1IF is set, and the CCIOF bit is set when CC1IF has been set before. If the CC1IE bit is set, then an interrupt is generated; if CC1DE is set, a DMA request is generated. An input capture event can be generated by software by way of writing the event generation register (R16_TIMx_SWEVGR).

### 10.3.2 Compare Output Mode

The compare output mode is one of the basic functions of the timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) agrees with the value of the compare capture register. the OCxM field (in R16_TIM2_CHCTLRx) and the CCxP bit (in R16_TIM2_CCER) determine whether the output is a definite high or low level or a level flip. The CCxIF bit is also set when a compare coherent event is generated. If the CCxIE bit is pre-set, an interrupt will be generated; if the CCxDE bit is pre-set, a DMA request will be generated.

To configure to compare output modes, proceed as follows:

1) Configure the clock source and auto-reload value of the core counter (CNT).
2) Set the count value to be compared to the comparison capture register (R16_TIM2_CHxCVR).
3) Set the CCxIE bit if an interrupt needs to be generated.
4) Keep OCxPE at 0 to disable the preload register for the compare capture register.
5) Setting the output mode, setting the OCxM field and the CCxP bit.
6) Enable the output, setting the CCxE bit.
7) Set the CEN bit to start the timer.

### 10.3.3 Forced Output Mode

The output pattern of the timer's compare capture channel can be forced by software to output a determined level without relying on comparison of the compare capture register's shadow register with the core counter.

This is done by setting OCxM to 100b, which forces OCxREF to low, or by setting OCxM to 101b, which forces OCxREF to high.

Note that by forcing OCxM to 100b or 101b, the comparison process between the internal main counter and the compare capture register is still going on, the corresponding flags are still set, and interrupts and DMA requests are still being generated.

### 10.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of PWM and is a special case of the input capture mode. The operation is the same as input capture mode except for the following differences: PWM occupies 2 compare capture channels and the input polarity of the two channels is set to opposite, one of the signals is set as trigger input and SMS is set to reset mode.

For example, to measure the period and frequency of the PWM wave input from TI1, the following operations are required.

1) Set TI1 (TI1FP1) to be the input of IC1 signal. Set CC1S to 01b.
2) Set TI1FP1 to rising edge active. Holding CC1P at 0.
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b.
4) Select TI1FP2 to set to falling edge active. Set CC2P to 1.
5) Select TI1FP1 as the source of the clock source. set TS to 101b.
6) Set the SMS to reset mode, i.e. 100b.
7) Enables input capture. CC1E and CC2E are set.

*Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIM3_CH1/TIM3_CH2 can be used in the PWM input mode.*

### 10.3.5 PWM Output Mode

PWM output mode is one of the basic functions of the timer. PWM output mode is most commonly used to determine the PWM frequency using the reload value and the duty cycle using the capture comparison register. Set 110b or 111b in the OCxM field to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit to enable the automatic reload of the preload register. The value of the preload register can only be sent to the shadow register when an update event occurs, so the UG bit needs to be set to initialize all registers before the core counter starts counting.

- Edge alignment

When edge alignment is used, the core counter counts up or down. In the case of PWM mode 1, when the value of the core counter is greater than the comparison capture register, OCxREF rises to high. When the value of the core counter is less than the comparison capture register (for example, when the core counter increases to the value of R16_TIM2_ATRLR and returns to all zeros), OCxREF drops to low.

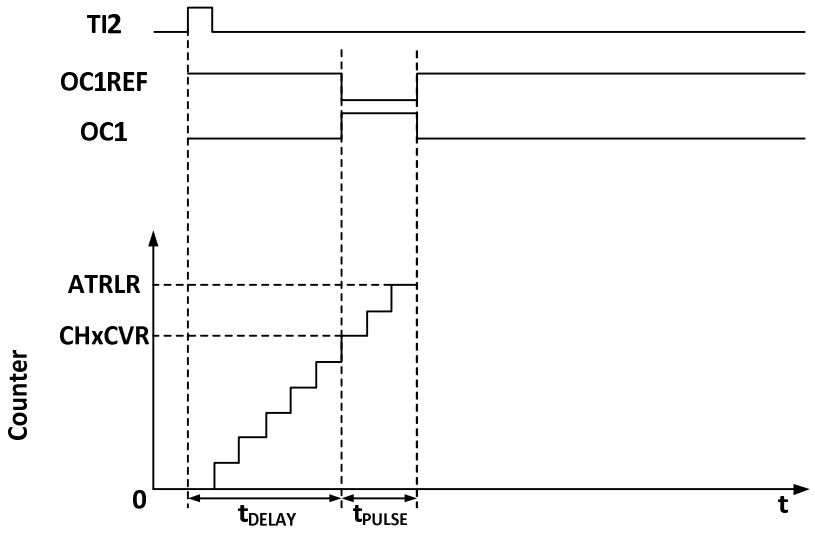
- Central alignment

When the central alignment mode is used, the core counter runs in the mode of alternately counting up and counting down, and OCxREF jumps up and down when the values of the core counter and the comparison capture register are consistent. However, the timing of setting the comparison mark is different under three central alignment modes. When using the central alignment mode, it is best to generate a software update flag (set UG bit) before starting the core counter.

### 10.3.6 Single Pulse Mode

The single pulse mode can respond to a specific event by generating a pulse after a delay, with programmable delay and pulse width. Setting the OPM bit stops the core counter when the next update event UEV is generated (Counter flips to 0).

Figure 10-4 Event generation and impulse response




As shown in Figure 10-4, a positive pulse of length Tpulse needs to be generated on OC1 after a delay Tdelay at the beginning of a rising edge detected on the TI2 input pin:

1) Set TI2 to trigger. Setting the CC2S field to 01b to map TI2FP2 to TI2; setting the CC2P bit to 0b to set TI2FP2

as rising edge detection; setting the TS field to 110b to set TI2FP2 as trigger source; setting the SMS field to 110b to set TI2FP2 to be used to start the counter.

2) Tdelay is defined by the Compare Capture Register and Tpulse is determined by the value of the Auto Reload Value Register and the Compare Capture Register.

### 10.3.7 Encoder mode

Encoder mode is a typical application of timer, which can be used to access the dual-phase output of encoder. The counting direction of core counter is synchronous with the direction of encoder shaft, and the core counter will be increased or decreased by one every pulse output by encoder. The steps of using the encoder are as follows: set the SMS field to 001b (counting only at the edge of TI2), 010b (counting only at the edge of TI1) or 011b (counting only at the edges of TI1 and TI2), connect the encoder to the input ends of comparative capture channels 1 and 2, and set the value of a reset value counter, which can be set larger. When in encoder mode, the comparison capture register, prescaler and repeat count register inside the timer all work normally. The following table shows the relationship between counting direction and encoder signal.

Table 10-1 Relationship between counting direction of timer encoder mode and encoder signal

<table>
<tr><th rowspan="2">Counting effective edges</th><th rowspan="2">The level of relative signals</th><th colspan="2">TI1FP1 signal edge</th><th colspan="2">TI2FP2 signal</th></tr>
<tr><th>Rising edge</th><th>Falling edge</th><th>Rising edge</th><th>Falling edge</th></tr>
<tr><td rowspan="2">Counting at TI1 edge only</td><td>high</td><td>Downward counting</td><td>Upward counting</td><td colspan="2" rowspan="2">No count</td></tr>
<tr><td>low</td><td>Upward counting</td><td>Downward counting</td></tr>
<tr><td rowspan="2">Counting at TI2 edge only</td><td>high</td><td colspan="2" rowspan="2">No count</td><td>Upward counting</td><td>Downward counting</td></tr>
<tr><td>low</td><td>Downward counting</td><td>Upward counting</td></tr>
<tr><td rowspan="2">Double edge counting at TI1 and TI2</td><td>high</td><td>Downward counting</td><td>Upward counting</td><td>Upward counting</td><td>Downward counting</td></tr>
<tr><td>low</td><td>Upward counting</td><td>Downward counting</td><td>Downward counting</td><td>Upward counting</td></tr>
</table>

### 10.3.8 Timer Synchronization Mode

Timer can output clock pulse (TRGO) and also receive input from other timers (ITRx). The source of ITRx of different timers (TRGO of other timers) is different. The internal trigger connection of timer is shown in Table 10-2.

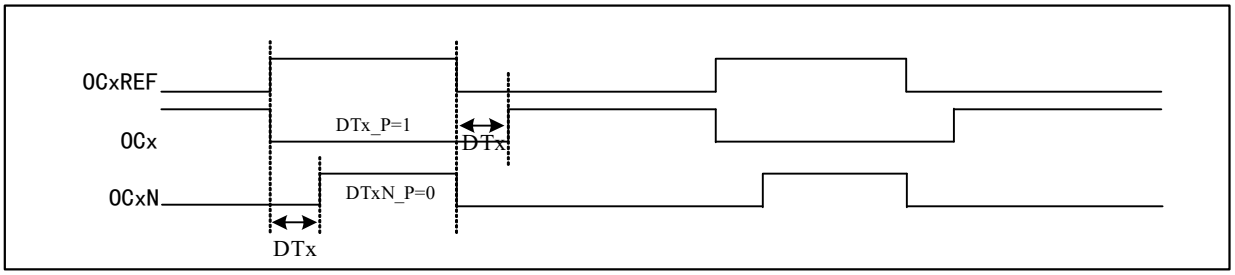
Table 10-2 GTPM internal trigger connection

| Slave timer | ITR0(TS=000) | ITR1(TS=001) | ITR2(TS=010) | ITR3(TS=011) |
|---|---|---|---|---|
| TIM2 | TIM1 | - | USB | - |
| TIM1 | TIM2 | | USB | - |

### 10.3.9 Complementary Outputs and Dead-time

TIM2 can output two complementary signals with dead zone (OCx and OCxN), and the polarity is controlled independently by DTx_P and DTxN_P bits, and the size of dead zone time can be set by setting DTx bits.

Figure 10-5 Complementary output and dead time




When complementary output, channel 3 and channel 4 serve as complementary output channel of channel 1 and channel 2.

The complementary or co-directional output of configurable dead time is multiplexed with the output channels of channels 3 and 4 by configuring bits OC1N_EN and OC2N_EN.

### 10.3.10 Dual Edge Capture Mode

The pulse can be measured by turning on the double-edge capture of the corresponding channel through the CAP_ED_CHx bit in the TIM2_CTLR2 register.

For example, channel 2 is used to capture the pulse width. In the configuration of capture function, the source of IC2 signal is selected (CC2S bit in TIM2_CHCTLR1 register is 11b), and the double-edge capture function of channel 2 is enabled (CAP_ED_CH2 bit in TIM2_CTLR2 register is 1). So far, the configuration of double edge capture has been completed.

The high and low level width values of the captured pulse can be read through the CH2CVR bit of the TIM2_CH2CVR register, and the level corresponding to the captured value can be indicated through the bit[16] of the TIM2_CH2CVR register by turning on the CAPLVL bit.

### 10.3.11 High Frequency Clock Counting Mode

Channels 1, 2 and corresponding complementary channels of Timer 2 support the use of PLL clock as the working clock of the core counter to generate PWM(OCxM bit can only select PWM mode 1 or PWM mode 2), and only the incremental counting mode is supported under this function; Bits DT2[3:0] of TIM2_DTCR register can be configured at the value of dead time on the falling edge of OCXREF, and DT1[3:0] can be configured at the value of dead time on the rising edge of OCXREF; DT_DIV bit turns on the function of dividing the dead time counting clock by 4, and extends the dead time.

### 10.3.12 Debug mode

When the system enters the debug mode, the timer can be controlled to continue running or stop according to the setting of DBG module.

## 10.4 Register Description

Table 10-3 TIM2-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_TIM2_CTLR1 | 0x40000000 | Control register 1 | 0x0000 |

| | | | |
|---|---|---|---|
| R16_TIM2_CTLR2 | 0x40000004 | Control register 2 | 0x0000 |
| R16_TIM2_SMCFGR | 0x40000008 | Slave mode control register | 0x0000 |
| R16_TIM2_DMAINTENR | 0x4000000C | DMA/ interrupt enable register | 0x0000 |
| R16_TIM2_INTFR | 0x40000010 | Interrupt status register | 0x0000 |
| R16_TIM2_SWEVGR | 0x40000014 | Event generation register | 0x0000 |
| R16_TIM2_CHCTLR1 | 0x40000018 | Compare/capture control register 1 | 0x0000 |
| R16_TIM2_CHCTLR2 | 0x4000001C | Compare/capture control register 2 | 0x0000 |
| R16_TIM2_CCER | 0x40000020 | Compare/capture enable register | 0x0000 |
| R16_TIM2_CNT | 0x40000024 | Counter of general-purpose timer | 0x0000 |
| R16_TIM2_PSC | 0x40000028 | Count clock prescaler | 0x0000 |
| R16_TIM2_ATRLR | 0x4000002C | Auto-reload value register | 0xFFFF |
| R32_TIM2_CH1CVR | 0x40000034 | Compare/capture register 1 | 0x00000000 |
| R32_TIM2_CH2CVR | 0x40000038 | Compare/capture register 2 | 0x00000000 |
| R32_TIM2_CH3CVR | 0x4000003C | Compare/capture register 3 | 0x00000000 |
| R32_TIM2_CH4CVR | 0x40000040 | Compare/capture register 4 | 0x00000000 |
| R16_TIM2_DTCR | 0x40000044 | Deadtime function configuration register | 0x0000 |
| R16_TIM2_DMACFGR | 0x40000048 | DMA control register | 0x0000 |
| R16_TIM2_DMAADR | 0x4000004C | DMA address register in continuous mode | 0x0000 |

### 10.4.1 Control Register 1 (TIM2_CTLR1)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | CAPLVL | RW | In double-edge capture mode, the capture level indication is enabled.<br>1: Enable the indication function;<br>0: Disable the indication function.<br>*Note: When enabled, [16] of CHxCVR indicates the level corresponding to the capture value.* | 0 |
| 14 | CAPOV | RW | Capture value mode configuration.<br>1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture;<br>0: The capture value is the actual counter value. | 0 |
| 13 | Reserved | RO | Reserved | 0 |
| 12 | CH2_PWMOUT_EN | RW | The PWM of timer channel 2 is enabled from the PC5 pin output:<br>1: Enable;<br>0: Disable.<br>*Note: (1) When the TIM2_RM bit in the AFIO_PCFR1 register is 10b, it is recommended to turn on this bit, and the rest are recommended to set to the default value. (2) When PC5 performs the TIM2_CH2 output pin, it only supports driving high level, and other cases are in high-impedance state.* | 0 |
| 13 | Reserved | RO | Reserved | 0 |
| [9:8] | CKD[1:0] | RW | These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the sampling clock used by the digital filter.<br>00: Tdts=Tck_int;<br>01: Tdts= 2xTck_int;<br>10: Tdts= 4xTck_int;<br>11： Reserved. | 0 |
| 7 | ARPE | RW | Auto-reload preload enable bit.<br>1: Enable the Auto-reload value register (ATRLR).<br>0: disabled the Auto-reload value register (ATRLR). | 0 |
| [6:5] | CMS[1:0] | RW | Central alignment mode selection: | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 00: Edge Alignment Mode. The counter counts up or down based on the direction bit (DIR);<br>01: Central alignment mode 1. Counters alternately count up and down. The output comparison interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts downward;<br>10: Central alignment mode 2. Counters alternately count up and down. The output comparison interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts upward;<br>11: Central alignment mode 3. Counters alternately count up and down. The output comparison interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set when the counter counts up and down.<br>*Note: When the counter is enabled (CEN=1), the transition from edge alignment mode to center alignment mode is not allowed.* | |
| 4 | DIR | RW | Counter direction:<br>1: The counter counting mode is down counting;<br>0: The counter counting mode is up counting.<br>*Note: This bit is invalid when the counter is configured in central alignment mode or encoder mode.* | 0 |
| 3 | OPM | RW | Single pulse mode.<br>1: The counter stops when the next update event (Clearing the CEN bit) occurs.<br>0: The counter does not stop when the next update event occurs. | 0 |
| 2 | URS | RW | Update request source, by which the software selects the source of the UEV event.<br>1: If an update interrupt or DMA request is enabled, only an update interrupt or DMA request is generated if the counter overflows/underflows.<br>0: If an update interrupt or DMA request is enabled, the update interrupt or DMA request is generated by any of the following events.<br>-Counter overflow/underflow<br>-Set the UG position<br>-Update generated from the mode controller | 0 |
| 1 | UDIS | RW | Disable updates, the software allows/disables the generation of UEV events via this bit.<br>1: UEV is disabled. no update event is generated and the registers (ATRLR, PSC, CHCTLRx) maintain their values. If the UG bitis set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized.<br>0: UEV is allowed. update (UEV) events are generated by any of the following events:<br>- Counter overflow/underflow<br>-Set the UG position<br>-Update generated from the mode controller registers with caches are loaded with their preloaded values. | 0 |
| 0 | CEN | RW | Enable the counter.<br>1: Enable the counter.<br>0: Disable the counter. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | *Note: The external clock, gated mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.* | |

### 10.4.2 Control Register 2 (TIM2_CTLR2)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:10] | Reserved | RO | Reserved | 0 |
| 9 | HSCK_MODE | RW | Counter clock selection bits. Use this function to enable the PLL clock:<br>1: The core counters of channels 1 and 2 use the PLL clock as the working clock;<br>0: Channels 1 and 2 maintain the original clock signal as the operating clock. | 0 |
| 8 | ADC_CAP | RW | Capture ADC module signal edge function enabled:<br>1: Alternate channel 1 double edge capture realizes the function, and the counter is cleared after capture;<br>0: The function is off, and the double-edge capture function of Channel 1 is normal. | 0 |
| 7 | TI1S | RW | TI1 selection:<br>1: The pins TIM2_CH1, TIM2_CH2 and TIM2_CH3 are XOR-connected to the input of TI1;<br>0: The TIM2_CH1 pin is directly connected to the TI1 input. | 0 |
| [6:4] | MMS[2:0] | RW | Master mode selection: These three bits are used to select the synchronization information (TRGO) sent to the slave timer in the master mode. The possible combinations are as follows:<br>000: Reset -UG bit is used as the trigger output (TRGO). If the reset is triggered by the input (the slave mode controller is in reset mode), the signal on TRGO will have a delay compared with the actual reset;<br>001: Enable–The counter enable signal CNT_EN is used as the trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or control to enable slave timers within a period of time. The counter enable signal is generated by the logical OR of the CEN control bit and the trigger input signal in the gating mode. When the counter enable signal is controlled by the trigger input, there will be a delay on TRGO unless the master/slave mode is selected (see the description of MSM bit in TIM2_SMCFGR register);<br>010: The update event is selected as the trigger input (TRGO). For example, the clock of a master timer can be used as a prescaler of a slave timer;<br>011: Comparison pulse: when a capture or a comparison is successful, when the CC1IF flag is to be set (even if it is already high), the trigger output sends a positive pulse (TRGO);<br>100: OC1REF signal is used as trigger output (TRGO);<br>101: OC2REF signal is used as trigger output (TRGO);<br>110: OC3REF signal is used as trigger output (TRGO);<br>111: OC4REF signal is used as trigger output (TRGO); | 0 |
| 3 | CCDS | RW | 1: When an update event occurs, send the DMA request of CHxCVR;<br>0: When CHxCVR occurs, a DMA request for | 0 |

| | | | | |
|---|---|---|---|---|
| | | | CHxCVR is generated. | |
| 2 | CAP_ED_CH4 | RW | Channel 4 dual edge capture function enable:<br>1: Channel 4 dual edge capture function enable;<br>0: Channel 4 dual edge capture function closes. | 0 |
| 1 | CAP_ED_CH3 | RW | Channel 3 dual edge capture function enable:<br>1: Channel 3 dual edge capture function enable;<br>0: Channel 3 dual edge capture function closes. | 0 |
| 0 | CAP_ED_CH2 | RW | Channel 2 dual edge capture function enable:<br>1: Channel 2 dual edge capture function enable;<br>0: Channel 2 dual edge capture function closes. | 0 |

### 10.4.3 Slave Mode Control Register (TIM2_SMCFGR)

Offset address: 0x08

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | ETP | RO | ETR triggers polarity selection, which selects whether to directly input ETR or invert the input ETR.<br>1: Invert ETR, the low level or the falling edge is valid;<br>0: ETR, the high level or the rising edge is valid. | 0 |
| 14 | ECE | RW | External Clock Mode 2 enables selection.<br>1: Enable external clock mode 2;<br>0: Disable external clock mode 2.<br>*Note 1: The slave mode can be used simultaneously with external clock mode 2: reset mode, gate mode and trigger mode; however, at this time TRGI cannot be connected to ETRF (TS bit cannot be 111b).*<br>*Note 2: When external clock mode 1 and external clock mode 2 are enabled at the same time, the input of the external clock is ETRF.* | 0 |
| [13:12] | ETPS[1:0] | RW | The frequency of the external trigger signal (ETRP) can be divided. The maximum frequency of this signal cannot exceed 1/4 of the TIM2CLK frequency. The frequency can be reduced through this domain.<br>00: Turn off pre-frequency division;<br>01: ETRP frequency divided by 2;<br>10: ETRP frequency divided by 4;<br>11: ETRP frequency is divided by 8. | 0 |
| [11:8] | ETF[3:0] | RW | External trigger filtering, in fact, the digital filter is an event counter, which uses a certain sampling frequency and will produce an output jump after recording n events.<br>0000: No filter, sampling with Fdts;<br>0001: Sampling frequency Fsampling=Fck_int, N=2;<br>0010: Sampling frequency Fsampling=Fck_int, N=4;<br>0011: Sampling frequency Fsampling=Fck_int, N=8;<br>0100: Sampling frequency Fsampling=Fdts/2, N=6;<br>0101: Sampling frequency Fsampling=Fdts/2,N=8;<br>0110: Sampling frequency Fsampling=Fdts/4, N=6;<br>0111: Sampling frequency Fsampling=Fdts/4, N=8;<br>1000: Sampling frequency Fsampling=Fdts/8, N=6<br>1001: Sampling frequency Fsampling=Fdts/8, N=8;<br>1010: Sampling frequency Fsampling=Fdts/16, N=5;<br>1011: Sampling frequency Fsampling=Fdts/16, N=6;<br>1100: Sampling frequency Fsampling=Fdts/16, N=8;<br>1101: Sampling frequency Fsampling=Fdts/32, N=5;<br>1110: Sampling frequency Fsampling=Fdts/32, N=6;<br>1111: Sampling frequency Fsampling=Fdts/32, N=8. | 0 |

| | | | | |
|---|---|---|---|---|
| 7 | MSM | RW | Master/slave mode selection.<br>1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is very useful when it is required to synchronize several timers to a single external event;<br>0: It doesn't work. | 0 |
| [6:4] | TS[2:0] | RW | Trigger select field, these 3 bits select the trigger input source used to synchronize the counter.<br>000: Internal trigger 0 (ITR0).<br>001: Internal trigger 1 (ITR1).<br>010: Internal trigger 2 (ITR2).<br>011: Internal trigger 3 (ITR3).<br>100: Edge detector of TI1 (TI1F_ED).<br>101: Filtered timer input 1 (TI1FP1).<br>110: Filtered timer input 2 (TI2FP2).<br>111: External trigger input (ETRF).<br>The above only changes when SMS is 0. | 0 |
| 3 | Reserved | RO | Reserved. | 0 |
| [2:0] | SMS[2:0] | RW | Input mode selection field. Selects the clock and trigger mode of the core counter.<br>000: Driven by the internal clock CK_INT.<br>001: Reserved.<br>010: Reserved.<br>011: Reserved.<br>100: Reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers.<br>101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled.<br>110: Trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled.<br>111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter. | 0 |

### 10.4.4 DMA/Interrupt Enable Register (TIM2_DMAINTENR)

Offset address: 0x0C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | Reserved | RO | Reserved | 0 |
| 14 | TDE | RW | Trigger DMA request Enable bits:<br>1: Allow DMA request to be triggered;<br>0: Disable DMA request to be triggered. | 0 |
| 13 | Reserved | RO | Reserved | 0 |
| 12 | CC4DE | RW | Compare/capture channel 4 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 4;<br>0: Disable the DMA request of compare capture channel 4. | 0 |
| 11 | CC3DE | RW | Compare/capture channel 3 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 3;<br>0: Disable the DMA request of compare capture channel 3. | 0 |

| | | | | |
|---|---|---|---|---|
| 10 | CC2DE | RW | Compare/capture channel 2 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 2;<br>0: Disable the DMA request of compare capture channel 2. | 0 |
| 9 | CC1DE | RW | Compare/capture channel 1 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 1;<br>0: Disable the DMA request of compare capture channel 1. | 0 |
| 8 | UDE | RW | Updated DMA request enable bits:<br>1: Allow updated DMA requests;<br>0: Disable updated DMA requests. | 0 |
| 7 | Reserved | RO | Reserved | 0 |
| 6 | TIE | RW | Trigger interrupt enable bits:<br>1: Enable trigger interrupt;<br>0: Disable triggering interrupt. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | CC4IE | RW | Compare capture channel 4 interrupt enable bit<br>1: Enable compare capture channel 4 interrupt;<br>0: Disable compare capture channel 4 interrupt. | 0 |
| 3 | CC3IE | RW | Compare capture channel 3 interrupt enable bit<br>1: Enable compare capture channel 3 interrupt;<br>0: Disable compare capture channel 3 interrupt. | 0 |
| 2 | CC2IE | RW | Compare capture channel 2 interrupt enable bit<br>1: Enable compare capture channel 2 interrupt;<br>0: Disable compare capture channel 2 interrupt. | 0 |
| 1 | CC1IE | RW | Compare capture channel 1 interrupt enable bit<br>1: Enable compare capture channel 1 interrupt;<br>0: Disable compare capture channel 1 interrupt. | 0 |
| 0 | UIE | RW | Update interrupt enable bit<br>1: Enable updated interrupt;<br>0: Disable updated interrupt. | 0 |

### 10.4.5 Interrupt Status Register (TIM2_INTFR)

Offset address: 0x10

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:13] | Reserved | RO | Reserved. | 0 |
| 12 | CC4OF | RW0 | Compare capture channel 4 over-capture flag bit | 0 |
| 11 | CC3OF | RW0 | Compare capture channel 3 over-capture flag bit | 0 |
| 10 | CC2OF | RW0 | Compare capture channel 2 over-capture flag bit | 0 |
| 9 | CC1OF | RW0 | Compare capture channel 1 over-capture flag bit is only used when the compare capture channel is configured for input capture mode. This flag is set by hardware and a software write of 0 clears this bit.<br>1: The state of CC1IF has been set when the counter value is captured to the capture compare register;<br>0: No over-capture generated. | 0 |
| [8:7] | Reserved | RO | Reserved. | 0 |
| 6 | TIF | RW0 | The trigger interrupt flag bit, which is set by hardware to this position when a trigger event occurs and cleared by software. Trigger events include the detection of a valid edge at the TRGI input when from a mode other than gated, or any edge in gated mode.<br>1: Trigger event generated; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0: No trigger event generated. | |
| 5 | Reserved | RO | Reserved. | 0 |
| 4 | CC4IF | RW0 | Compare capture channel 4 interrupt flag bit. | 0 |
| 3 | CC3IF | RW0 | Compare capture channel 3 interrupt flag bit. | 0 |
| 2 | CC2IF | RW0 | Compare capture channel 2 interrupt flag bit. | 0 |
| 1 | CC1IF | RW0 | Compare capture channel 1 interrupt flag bit.<br>If the compare capture channel is configured in output mode, this bit is set by hardware when the counter value matches the compare value, except in centrosymmetric mode. This bit is cleared by software.<br>1: The value of the core counter matches the value of compare capture register 1; 0: no match occurs.<br>If compare capture channel 1 is configured in input mode, this bit is set by hardware when a capture event occurs and it is cleared by software or by reading the compare capture register.<br>1: The counter value has been captured compare capture register 1;<br>0: No input capture is generated. | 0 |
| 0 | UIF | RW0 | Update interrupt flag bit, this bit is set by hardware when an update event is generated and is cleared by software.<br>1: Update interrupt generated;<br>0: No update event is generated.<br>An update event will be generated in the following cases:<br>If UDIS=0, when the repeat counter value overflows or underflows;<br>If URS=0, UDIS=0, when the UG bit is set, or when the counter core counter is reinitialized by software;<br>If URS=0, UDIS=0, when the counter CNT is reinitialized by a trigger event; | 0 |

### 10.4.6 Event Generation Register (TIM2_SWEVGR)

Offset address: 0x14

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:7] | Reserved | RO | Reserved | 0 |
| 6 | TG | WO | The trigger event generation bit, which is set by software and cleared by hardware, is used to generate a trigger event.<br>1: Generate a trigger event, TIF is set, and the corresponding interrupts and DMAs are generated if enabled.<br>0: No action. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | CC4G | WO | Compare capture event generation bit 4. Generate compare capture event 4. | 0 |
| 3 | CC3G | WO | Compare capture event generation bit 3. Generate compare capture event 3. | 0 |
| 2 | CC2G | WO | Compare capture event generation bit 2. Generate compare capture event 2. | 0 |
| 1 | CC1G | WO | Compare capture event generation bit 1. Generate compare capture event 1. This bit is set by software and cleared by hardware.<br>It is used to generate a compare capture event. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Generate a compare capture event on compare capture channel 1.<br>If compare capture channel 1 is configured as output: set the CC1IF bit. Generate the corresponding interrupts and DMAs if they are enabled.<br>If compare capture channel 1 is configured as input: the current core counter value is captured to compare capture register 1; set the CC1IF bit and generate the corresponding interrupts and DMAs if they are enabled. If CC1IF is already set, set the CC1OF bit.<br>0: No action. | |
| 0 | UG | WO | Update event generation bit to generate an update event. This bit is set by software and is automatically cleared by hardware.<br>1: Initialize the counter and generate an update event.<br>0: No action.<br>*Note: The prescaler counter is also cleared to zero, but the prescaler factor remains unchanged. The core counter is cleared if in centrosymmetric mode or incremental counting mode; if in decremental counting mode, the core counter takes the value of the reload value register.* | 0 |

### 10.4.7 Compare/Capture Control Register 1 (TIM2_CHCTLR1)

Offset address: 0x18

The channel can be used in input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register serve different purposes in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC2CE | OC2M[2:0] | | | OC2PE | OC2FE | CC2S[1:0] | | OC1CE | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |
| IC2F[3:0] | | | | IC2PSC[1:0] | | | | IC1F[3:0] | | | | IC1PSC[1:0] | | | |

Compare mode (pin direction is output).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | OC2CE | RW | Compare capture channel 2 clear enable bit.<br>1: Clear the OC2REF bit zero once the ETRF input is detected high.<br>0: OC2REF is not affected by the ETRF input. | 0 |
| [14:12] | OC2M[2:0] | RW | Compare the Capture Channel 2 mode setting field.<br>The 3 bits define the action of the output reference signal OC2REF, which determines the values of OC2, OC2N. OC2REF is active high, while the active levels of OC2 and OC2N depend on the CC2P, CC2NP bits.<br>000: Freeze. Comparison of the value of the capture register with the value of the comparison between the core counters does not work for OC1REF.<br>001: Force to set to valid level. Forcing OC1REF high when the core counter has the same value as the comparison capture register 2.<br>010: Force to set to invalid level. Forcing OC1REF low when the value of the core counter is the same as the comparison capture register 2. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 011: Flip. Flips the level of OC1REF when the core counter is the same as the value of compare capture register 2.<br>100: Forced to invalid level. Forces OC1REF to low.<br>101: Force to valid level. Force OC1REF to high.<br>110: PWM mode 1: When counting up, once the core counter is less than the value of the comparison capture register, channel 2 is an effective level, otherwise it is an invalid level; When counting down, once the core counter is greater than the value of the comparison capture register, channel 2 is at an invalid level (OC2REF=0), otherwise it is at an effective level (OC2REF = 1);<br>111: PWM mode 2: When counting up, once the core counter is less than the value of the comparison capture register, channel 2 is at an invalid level, otherwise it is at an effective level; When counting down, once the core counter is greater than the value of the comparison capture register, Channel 2 is active (OC2REF=1), otherwise it is inactive (OC2REF=0).<br>*Note: This bit cannot be modified once the LOCK level is set to 3 and CC1S=00b. In PWM mode 1 or PWM mode 2, the OC1REF level is changed only when the comparison result is changed or when switching from freeze mode to PWM mode in the output comparison mode.* | |
| 11 | OC2PE | RW | Compare Capture Register 2 preload enable bit.<br>1: Turn on the preload function of the compare capture register 2, the read and write operations operate only on the preload register, and the preload value of the compare capture register 2 is loaded into the current shadow register when the update event comes.<br>0: Disable the preload function of compare capture register 2. The compare capture register 2 can be written at any time, and the newly written value takes effect immediately.<br>*Note: Once the LOCK level is set to 3 and CC2S=00, this bit cannot be modified. PWM mode can be used only in single pulse mode (OPM=1) without confirming the pre-load register, otherwise its action is not determined.* | 0 |
| 10 | OC2FE | RW | Compare Capture Channel 2 fast enable bit, this bit is used to speed up the response of the compare capture channel output to trigger input events.<br>1: The active edge of the input to the flipflop acts as if a comparison match has occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the valid edge of the sample trigger and the output of the compare capture channel 2 is reduced to 3 clock cycles.<br>0: Based on the value of the counter and compare capture register 1, compare capture channel 2 operates normally, even if the flip-flop is open. The minimum delay to activate the compare capture channel 2 output is 5 clock cycles when the input of the flipflop has a valid edge.<br>OC2FE only works when the channel is configured to | 0 |

| | | | | |
|---|---|---|---|---|
| | | | PWM1 or PWM2 mode. | |
| [9:8] | CC2S[1:0] | RW | Compare capture channel 2 input selection fields.<br>00: comparison capture channel 2 is configured as an output.<br>01: comparison capture channel 2 is configured as an input and IC2 is mapped on TI2.<br>10: comparison capture channel 2 is configured as an input and IC2 is mapped onTI1.<br>11: Compare Capture Channel 2 is configured as an input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (By the TS bit).<br>*Note: Compare capture channel 2 is writable only when the channel is off (When CC2E is zero).* | 0 |
| 7 | OC1CE | RW | Compare capture channel 1 clear enable bit. | 0 |
| [6:4] | OC1M[2:0] | RW | Compare capture channel 1 mode setting field. | 0 |
| 3 | OC1PE | RW | Compare capture register 1 preload enable bit. | 0 |
| 2 | OC1FE | RW | Compare capture channel 1 fast enable bit. | 0 |
| [1:0] | CC1S[1:0] | RW | Compare capture channel 1 input selection fields. | 0 |

Capture mode (pin direction is input).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | IC2F[3:0] | RW | The input capture filter 2 configuration field, these bits set the sampling frequency of the TI1 input and the digital filter length. The digital filter consists of an event counter, which records N events and then generates a jump in the output.<br>0000: No filter, sampled at fDTS.<br>1000: Sampling frequency Fsampling = Fdts/8, N = 6.<br>0001: Sampling frequency Fsampling=Fck_int, N=2.<br>1001: Sampling frequency Fsampling = Fdts/8, N = 8.<br>0010: Sampling frequency Fsampling=Fck_int, N=4.<br>1010: Sampling frequency Fsampling = Fdts/16, N = 5.<br>0011: Sampling frequency Fsampling=f=Fck_int, N=8.<br>1011: Sampling frequency Fsampling = Fdts/16, N = 6.<br>0100: Sampling frequency Fsampling = Fdts/2, N = 6.<br>1100: Sampling frequency Fsampling = Fdts/16, N = 8.<br>0101: Sampling frequency Fsampling = Fdts/2, N = 8.<br>1101: Sampling frequency Fsampling = Fdts/32, N = 5.<br>0110: Sampling frequency Fsampling = Fdts/4, N = 6.<br>1110: Sampling frequency Fsampling = Fdts/32, N = 6.<br>0111: Sampling frequency Fsampling = Fdts/4, N = 8.<br>1111: Sampling frequency Fsampling=Fdts/32, N=8. | 0 |
| [11:10] | IC2PSC[1:0] | RW | Compare capture channel 2 prescaler configuration field, these 2 bits define the prescaler coefficient for compare capture channel 2. Once CC1E = 0, the prescaler is reset.<br>00: without prescaler, one capture is triggered for each edge detected on the capture input.<br>01: Capture triggered every 2 events.<br>10: Capture triggered every 4 events.<br>11: Capture is triggered every 8 events. | 0 |
| [9:8] | CC2S[1:0] | RW | Compare the capture channel 2 input selection field, these 2 bits define the direction of the channel (Input/output), and the selection of the input pin.<br>00: Comparative capture channel 1 channel is configured as an output. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 01: Comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI1.<br>10: Comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI2.<br>11: The compare capture channel 1 channel is configured as an input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (By the TS bit).<br>*Note: CC1S is writable only when the channel is off (CC1E is 0).* | |
| [7:4] | IC1F[3:0] | RW | Input capture filter 1 configuration field. | 0 |
| [3:2] | IC1PSC[1:0] | RW | Compare capture channel 1 prescaler configuration field. | 0 |
| [1:0] | CC1S[1:0] | RW | Compare capture channel 1 input selection fields. | 0 |

### 10.4.8 Compare/Capture Control Register 2 (TIM2_CHCTLR2)

Offset address: 0x1C

The channel can be used in input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register serve different purposes in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC4CE | OC4M[2:0] | | | OC4PE | OC4FE | CC4S[1:0] | | OC3CE | OC3M[2:0] | | | OC3PE | OC3FE | CC3S[1:0] | |
| IC4F[3:0] | | | | IC4PSC[1:0] | | | | IC3F[3:0] | | | | IC3PSC[1:0] | | | |

Compare mode (Pin direction is output).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | OC4CE | RW | Compare capture channel 4 clear enable bit. | 0 |
| [14:12] | OC4M[2:0] | RW | Compare capture channel 4 mode setting field. | 0 |
| 11 | OC4PE | RW | Compare capture register 4 preload enable bit. | 0 |
| 10 | OC4FE | RW | Compare capture channel 4 fast enable bit. | 0 |
| [9:8] | CC4S[1:0] | RW | Compare capture channel 4 input selection fields. | 0 |
| 7 | OC3CE | RW | Compare capture channel 3 clear enable bit. | 0 |
| [6:4] | OC3M[2:0] | RW | Compare capture channel 3 mode setting field. | 0 |
| 3 | OC3PE | RW | Compare capture register 3 preload enable bit. | 0 |
| 2 | OC3FE | RW | Compare capture channel 3 fast enable bit. | 0 |
| [1:0] | CC3S[1:0] | RW | Compare capture channel 3 input selection fields. | 0 |

Capture mode (pin direction is input).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | IC4F[3:0] | RW | Input capture filter 4 configuration field. | 0 |
| [11:10] | IC4PSC[1:0] | RW | Compare capture channel 4 prescaler configuration field. | 0 |
| [9:8] | CC4S[1:0] | RW | Compare capture channel 4 input selection fields. | 0 |
| [7:4] | IC3F[3:0] | RW | Input capture filter 3 configuration field. | 0 |
| [3:2] | IC3PSC[1:0] | RW | Compare capture channel 3 prescaler configuration fields. | 0 |
| [1:0] | CC3S[1:0] | RW | Compare capture channel 3 input selection fields. | 0 |

### 10.4.9 Compare/Capture Enable Register (TIM2_CCER)

Offset address: 0x20

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:14] | Reserved | RO | Reserved | 0 |
| 13 | CC4P | RW | Compare capture channel 4 output polarity setting bit. | 0 |
| 12 | CC4E | RW | Compare capture channel 4 output enable bit. | 0 |
| [11:10] | Reserved | RO | Reserved | 0 |
| 10 | CC3NE | RW | Compare capture channel 3 complementary output enable bits. | 0 |
| 9 | CC3P | RW | Compare capture channel 3 output polarity setting bit. | 0 |
| 8 | CC3E | RW | Compare capture channel 3 output enable bit. | 0 |
| [7:6] | Reserved | RO | Reserved | 0 |
| 6 | CC2NE | RW | Compare capture channel 2 complementary output enable bits. | 0 |
| 5 | CC2P | RW | Compare capture channel 2 output polarity setting bit. | 0 |
| 4 | CC2E | RW | Compare capture channel 2 output enable bit. | 0 |
| [3:2] | Reserved | RO | Reserved | 0 |
| 1 | CC1P | RW | Compare capture channel 1 output polarity setting bit.<br>CC1 channel is configured as output:<br>1: OC1 is active low;<br>0: OC1 is active high.<br>CC1 channel is configured as input:<br>This bit selects whether IC1 or the inverted signal of IC1 is used as the trigger or capture signal.<br>1: Inversion: Capture occurs at the falling edge of IC1; When used as an external trigger, IC1 is inverted;<br>0: Non-inverting: capture occurs at the rising edge of IC1; When used as an external trigger, IC1 does not invert. | 0 |
| 0 | CC1E | RW | Compare capture channel 1 output enable bit.<br>CC1 channel is configured as output:<br>1: On. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE.<br>0: Off. OC1 prohibits output, so the output level of OC1 depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits.<br>CC1 channel is configured as input:<br>This bit determines whether the value of the counter can be captured in the TIM1_CH1CVR register.<br>1: Capture enabled;<br>0: Capture disabled. | 0 |

### 10.4.10 Counter for Advanced-control Timer (TIM2_CNT)

Offset address: 0x24

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CNT[15:0] | RW | The real-time value of the timer's counter. | 0 |

### 10.4.11 Counting Clock Prescaler (TIM2_PSC)

Offset address: 0x28

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | PSC[15:0] | RW | The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the divider/(PSC+1). | 0 |

### 10.4.12 Auto-reload Value Register (TIM2_ATRLR)

Offset address: 0x2C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | ARR[15:0] | RW | The value of ATRLR[15:0] will be loaded into the counter. Please read section 10.2.4 when ATRLR is activated and updated. When ATRLR is empty, the counter stops. | 0xFFFF |

### 10.4.13 Compare/Capture Register 1 (TIM2_CH1CVR)

Offset address: 0x34

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL1 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH1CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL1 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH1CVR[15:0] | RW | Compare/capture the value of register channel 1. | 0 |

### 10.4.14 Compare/Capture Register 2 (TIM2_CH2CVR)

Offset address: 0x38

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL2 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH2CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL2 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH2CVR[15:0] | RW | Compare/capture the value of register channel 2. | 0 |

### 10.4.15 Compare/Capture Register 3 (TIM2_CH3CVR)

Offset address: 0x38

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL3 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH3CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL3 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH3CVR[15:0] | RW | Compare/capture the value of register channel 3. | 0 |

### 10.4.16 Compare/Capture Register 4 (TIM2_CH4CVR)

Offset address: 0x40

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH4CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL3 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH4CVR[15:0] | RW | Compare/capture the value of register channel 4. | 0 |

### 10.4.17 Deadtime Function Configuration Register (TIM2_DTCR)

Offset address: 0x44

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | DT2[3:0] | RW | Comparison register channel 1 and comparison register channel 2 are the values of dead time duration at the falling edge of OCXREF:<br>0000: Deadtime is 1*T;<br>0001: Deadtime is 2*T;<br>…<br>1110: Deadtime is 15*T;<br>1111: Deadtime is 16*T.<br>*Note: The width of dead zone must be less than the width of effective level.* | 0 |
| [11:8] | DT1[3:0] | RW | Comparison register channel 1, comparison register channel 2 in OCXREF rising edge dead time duration value:<br>0000: Deadtime is 1*T;<br>0001: Deadtime is 2*T;<br>…<br>1110: Deadtime is 15*T;<br>1111: Deadtime is 16*T.<br>*Note: The width of dead zone must be less than the width of effective level.* | 0 |
| 7 | Reserved | RO | Reserved | 0 |
| 6 | DT_DIV | RW | Dead time counting clock frequency division enable:<br>1: Clock divided by 4;<br>0: Clock does not divide. | 0 |
| 5 | DT2N_P | RW | Complementary output function-channel 2 complementary channel output polarity setting bit:<br>1: Active low;<br>0: Active high. | 0 |
| 4 | DT2_P | RW | Complementary output function-channel 2 output polarity setting bit:<br>1: Active low;<br>0: Active high. | 0 |
| 3 | DT1N_P | RW | Complementary output function-channel 1 | 0 |

| | | | | |
|---|---|---|---|---|
| | | | complementary channel output polarity setting bit:<br>1: Active low;<br>0: Active high. | |
| 2 | DT1_P | RW | Complementary output function-channel 1 output polarity setting bit:<br>1: Active low;<br>0: Active high. | 0 |
| 1 | OC2N_EN | RW | Complementary output function - Channel 2 and complementary channel output enable bits (Channel 4 is the complementary channel of channel 2):<br>1: The dead time function of channel 2 is enabled, complementary channel multiplexing channel 4;<br>0: The dead time function of channel 2 is turned off. | 0 |
| 0 | OC1N_EN | RW | Complementary output function - Channel 1 and complementary channel output enable bits (Channel 3 is the complementary channel of channel 1):<br>1: The dead time function of channel 1 is enabled, complementary channel multiplexing channel 3;<br>0: The dead time function of channel 1 is turned off. | 0 |

### 10.4.18 DMA Control Register (TIM2_DMACFGR)

Offset address: 0x48

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:13] | Reserved | RO | Reserved | 0 |
| [12:8] | DBL[4:0] | RW | The length of the DMA continuous transmission, the actual value of which is the value of this field + 1. | 0 |
| [7:5] | Reserved | RO | Reserved | 0 |
| [4:0] | DBA[4:0] | RW | These bits define the offset of the DMA in continuous mode from the address where control register 1 is located. | 0 |

### 10.4.19 DMA Address Register for Continuous Mode (TIM2_DMAADR)

Offset address: 0x4C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | DMAB [15:0] | RW | The address of the DMA in continuous mode. | 0 |

# Chapter 11 Streamlined Timer (SLTM)

The Streamlined Timer module contains a 16-bit automatic reinstallation timer TIM3, which is used to measure the pulse width or generate pulses, PWM waves of specific frequency, etc. Can be used in fields such as automation control and power supply.

## 11.1 Main Features

The main features of the streamlined timer include:

- 16-bit auto-reload counter, supports incremental counting mode
- 16-bit prescaler, the frequency division coefficient is dynamically adjustable from 1 to 65536
- Support 2 independent compare channels
- Each comparison and capture channel supports multiple operating modes, such as input capture, output comparison, PWM generation and single pulse output
- Support external signal control timer
- Support the use of DMA in multiple modes
- Supports cascading and synchronization between timers
- Supports simple dead time control

## 11.2 Principle and Structure

Figure 11-1 Block diagram of the structure of the streamlined timer

### 11.2.1 Overview

As shown in Figure 11-1, the structure of the streamlined timer can be roughly divided into three parts, namely the input clock part, the core counter part and the compare channel part.

The clock of the streamlined timer can come from the HB bus clock (CK_INT), from other timers (ITRx) with clock output function, and from the input terminal of the compare capture channel (TIM3_CHx). These input clock signals are then used to become CK_PSC clocks after various set filtering and frequency division operations, and are output to the core counter part.

The core of the streamlined timer is a 16-bit counter (CNT). CK_PSC input directly to CNT, CNT without frequency division supports incremental counting mode, decremental counting mode and incremental or decremental counting

mode, and has an automatic reload value register (ATRLR) to reload initialization values for CNT at the end of each counting cycle.

The streamlined timer has 2 sets of comparison capture channels, each of which can input pulses from the exclusive pins or output waveforms to the pins, that is, the compare capture channel supports input and output modes. The inputs of each channel of the compare capture register support filtering, frequency division, edge detection and other operations, and support mutual triggering between channels, and can also provide clocks for the core counter CNT. Each compare capture channel has a set of compare capture registers (CHxCVR), which supports comparison with the main counter (CNT) and output pulses.

### 11.2.2 Difference between Streamlined Timer and Advanced Timer

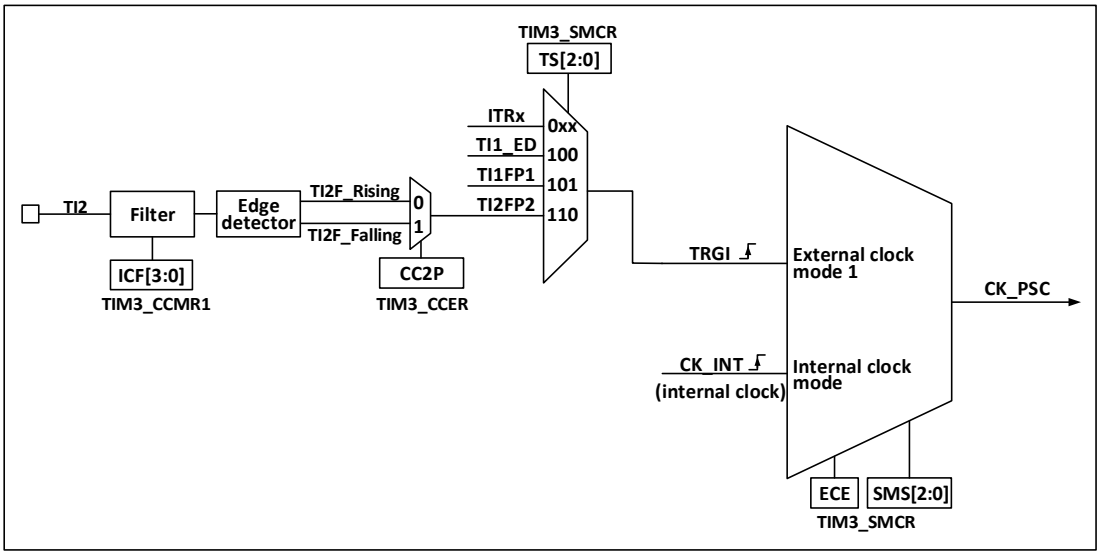
Compared to advanced timers, the streamlined timer lacks the following features:

1) The streamlined timer lacks a repeat count register that counts the counting cycles of the core counter.
2) The streamlined timer has no brake signal mechanism.
3) The streamlined timer has no encoder mode.
4) The streamlined timer 16-bit auto-reload counter does not support decrement counting mode and increment/decrement counting mode
5) The streamlined timer does not support external clock pin (ETR) input clock.

### 11.2.3 Clock Input

This section discusses the source of CK_PSC. The clock source part of the overall structure block diagram of the streamlined timer is intercepted here.

Figure 11-2 Streamlined timer CK_PSC source block diagram




The optional input clocks can be divided into 3 categories.

1) Internal HB clock input route: CK_INT
2) Route from the compare capture channel Pin (TIM3_CHx):
3) Input from other internal timers: ITRx.

By determining the input pulse selection of the SMS from which the CK_PSC comes from, the actual operation can be divided into 2 categories:

1) Select internal clock source (CK_INT)
2) External clock source mode 1

The 3 clock sources mentioned above can be selected by these 2 operations.

#### 11.2.3.1 Internal Clock Source (CK_INT)

If the compact timer is started when the SMS domain is kept at 000b, the internal clock source (CK_INT) is selected as the clock. At this point, CK_INT is CK_PSC.

#### 11.2.3.2 External Clock Source Mode 1

If the SMS domain is set to 111b, external clock source mode 1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source of CK_PSC. It is worth noting that the user also needs to configure the TS domain to select the source of TRGI. The following pulses can be selected as the clock source in the TS domain:

1) Internal trigger (ITRx, x is 0, 1, 2, 3);
2) Signal after compare capture channel 1 passes through edge detector (TI1F_ED);
3) Signal of compare capture channel TI1FP1, TI2FP2.

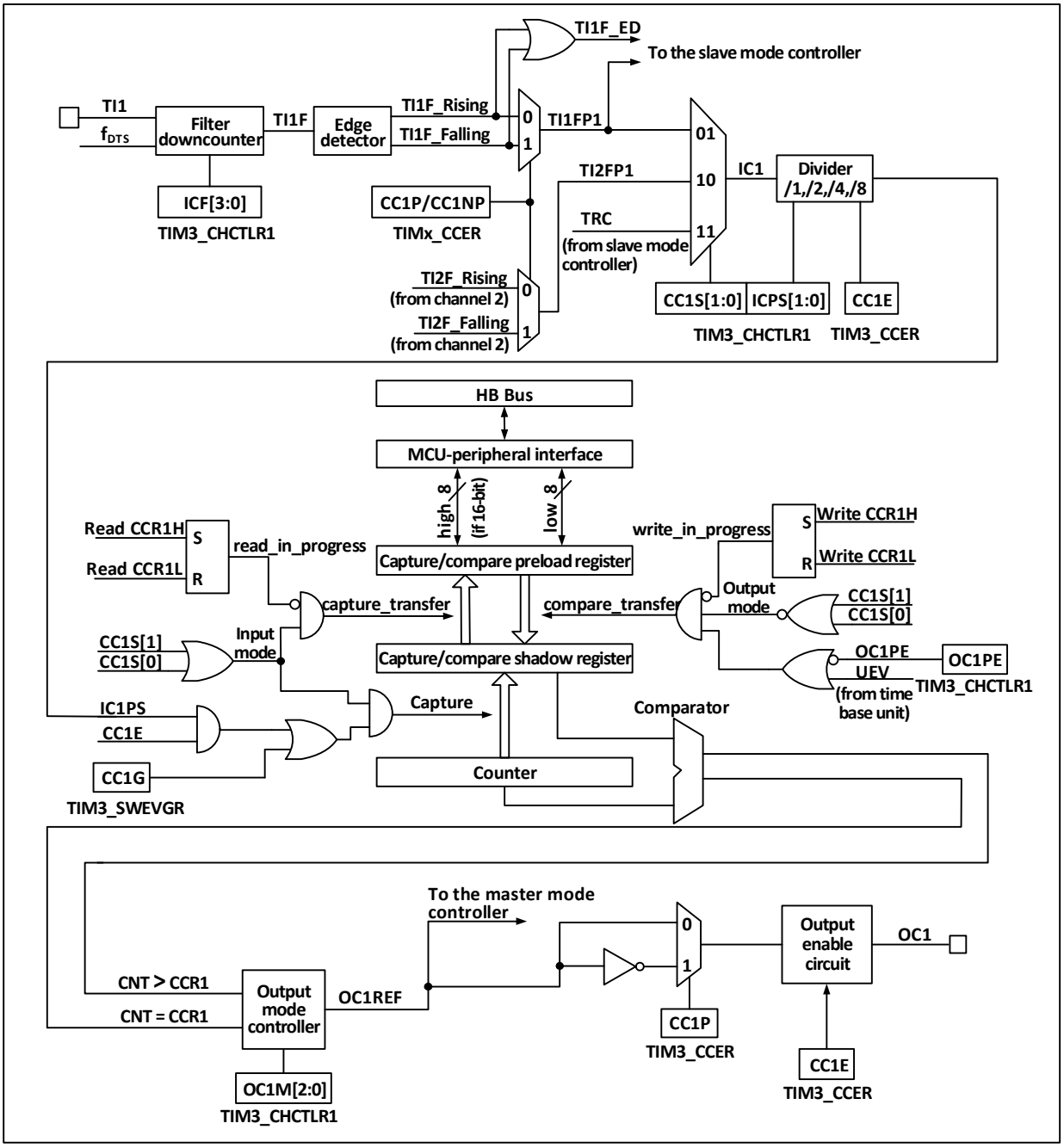
### 11.2.4 Counters and Peripherals

C CK_PSC input is divided by prescaler (PSC). PSC is 16 bits, and the actual frequency division coefficient is equivalent to the value of R16_TIM3_PSC +1. CK_PSC will become CK_INT after PSC. Changing the value of R16_TIM1_PSC will not take effect in real time, but will be updated to PSC after the update event. Update events include UG bit clearing and resetting.

### 11.2.5 Compare Capture Channel

The compare capture channel is the core of the timer to realize complex functions, and its core is the comparison capture register, supplemented by digital filtering, frequency division and channel multiplexing in the peripheral input part, comparator and output control in the output part. The structural block diagram of the comparison capture channel is shown in Figure 113.

Figure 11-3 Block diagram of the structure of the compare capture channel



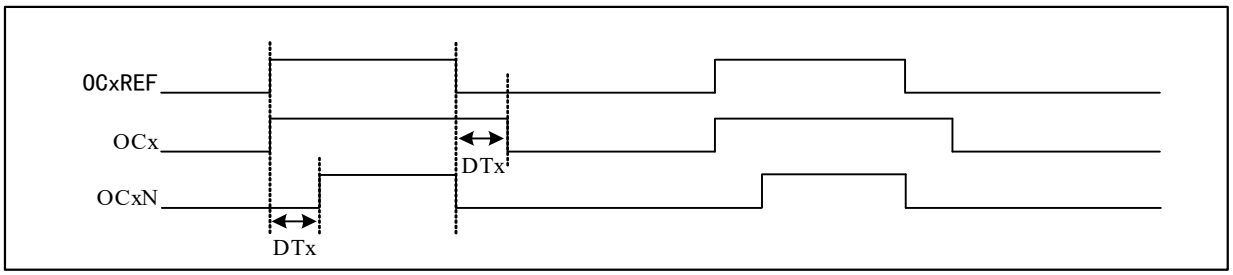

The signal comes in from the channel x pin and can be selected as TIx (TI1 can come from more than just CH1, see Block Diagram 11-1 of the timer), TI1 goes through the filter (ICF[3:0]) to generate TI1F, which is then divided by the edge detector into TI1F_Rising and TI1F_Falling, and these two signals are then selected (CC1P) to generate TI1FP1, TI1FP1 and TI2FP1 from channel 2 are sent together to CC1S to be selected as IC1, which is divided by ICPS and sent to the compare capture register.

The compare capture register consists of a preload register and a shadow register, and only the preload register is operated during reading and writing. In capture mode, capture occurs on the shadow register and then copied to the preloaded register; In compare mode, the contents of the preloaded register are copied into the shadow register, and then the contents of the shadow register are compared with the core counter (CNT).

### 11.2.6 Complementary Output and Dead-time

Generally, 2 channels have two output pins, which can output 2 complementary signals with dead time (OCx and OCxN). The polarity is controlled independently by COxP and COxNP bits, and the size of dead time can be set by setting DTx bits.

Figure 11-4 Complementary output and dead time




## 11.3 Function and Implementation

The realization of the complex functions of the streamlined timer is realized by operating the compare capture channel, clock input circuit, counter and peripheral components of the timer. The clock input of the timer can come from multiple clock sources including the input of the comparison capture channel. The selection of the compare capture register channel and clock source directly determines its function. The compare capture channel is bidirectional and can work in input and output modes.

### 11.3.1 Input Capture Mode

Input capture mode is one of the basic functions of timer. The principle of input capture mode is that when a certain edge on the ICxPS signal is detected, a capture event will be generated, and the current value of the counter will be latched into the comparison capture register (R16_TIM3_CHCTLRx). When a capture event occurs, CCxIF (in R16_TIM3_INTFR) is set, and if an interrupt or DMA is enabled, a corresponding interrupt or DMA will be generated. If CCxIF has been set when the capture event occurs, the CCxOF bit will be set. CCxIF can be cleared by software or by hardware by reading the comparison capture register. CCxOF is cleared by software.

Take channel 1 as an example to illustrate the steps of using input capture mode.

1) Configure CCxS domain and select the source of ICx signal. For example, if it is set to 10b, TI1FP1 is selected as the source of IC1, and the default setting cannot be used. The default setting of CCxS domain is to make the comparison capture module as the output channel.
2) Configure ICxF domain and set the digital filter of TI signal. The digital filter will sample a certain number of times at a certain frequency, and then output a jump. The sampling frequency and times are determined by ICxF;
3) Configure the CCxP bit and set the polarity of TIxFPx. For example, keep CC1P bit low and select rising edge transition;
4) Configure the ICxPS domain and set the ICx signal as the frequency division coefficient between ICxPS. For example, keep ICxPS at 00b without frequency division;
5) Configure the CCxE bit to allow capturing the value of the core counter (CNT) into the comparison capture register. Set CC1E bit;
6) Configure the CCxIE and CCxDE bits as needed to decide whether to enable interrupts or DMA.

At this point, the comparison capture channel has been configured.

When a captured pulse is input to TI1, the value of the core counter (CNT) will be recorded in the comparison capture register, and CC1IF will be set. When CC1IF has been set before, CCIOF bit will also be set. If CC1IE is set, an interrupt will be generated; If CC1DE is set, a DMA request will be generated. An input capture event can be generated by software by writing the event generation register (R16_TIM3_SWEVGR).

### 11.3.2 Compare Output Mode

Compare output mode is one of the basic functions of timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) is consistent with the value of the compare capture register. The OCxM field (in R16_TIM3_CHCTLRx) and the CCxP bit (in R16_TIM3_CCER) determine whether the output is a certain high-low level or a level inversion. CCxIF bit will be set when a relatively consistent event is generated, and if CCxIE bit is set in advance, an interrupt will be generated; If the CCxDE bit is set in advance, a DMA request will be generated.

The steps to configure to compare output modes are as follows:

1) Configure the clock source and automatic reload values of the core counter (CNT)
2) Set the count value to be compared to the compare register (R16_TIM3_CHxCVR)
3) If an interrupt is needed, set the CCxIE bit;
4) Keep OCxPE to 0 and disable preloaded registers that compare the obtained registers
5) Set output mode, set OCxM field and CCxP bit;
6) Enable output and set CCxE bit;
7) Set the CEN bit to start the timer.

### 11.3.3 Forced Output Mode

The output mode of the comparE capture channel of the timer can be forced to output a certain level by software, without relying on the comparison between the shadow register of the comparE capture register and the core counter.

The specific way is to set OCxM to 100b, that is, to set OCxREF to low forcibly; Or set OCxM to 101b, that is, force OCxREF to be set high.

It should be noted that when OCxM is forcibly set to 100b or 101b, the comparison process between the internal master counter and the comparison capture register is still in progress, the corresponding flag bit is still set, and interrupts and DMA requests are still generated.

### 11.3.4 PWM Input Mode

PWM input mode is used to measure the duty cycle and frequency of PWM, which is a special case of input capture mode. Except for the following differences, the operation and input capture mode are the same: PWM occupies 2 compare capture channels, and the input polarities of the 2 channels are set to be opposite, one of the signals is set to trigger input, and SMS is set to reset mode.

For example, to measure the period and frequency of PWM wave input from TI1, the following operations are required:

1) Set TI1(TI1FP1) as the input of IC1 signal. Set CC1S to 01b;
2) Make TI1FP1 valid as the rising edge. Keep CC1P at 0;
3) Set TI1(TI1FP2) as the input of IC2 signal. Set CC2S to 10b;
4) Select TI1FP2 as the effective falling edge. Set CC2P to 1;

5) The source of the clock source is TI1FP1. Set TS to 101b;
6) Set SMS to reset mode, that is, 100b;
7) Can input capture. CC1E and CC2E are set.

*Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIM3_CH1/TIM3_CH2 can be used in the PWM input mode.*

### 11.3.5 PWM Output Mode

PWM output mode is one of the basic functions of timer. The most common method of PWM output mode is to determine PWM frequency by reloading value and determine duty cycle by using capture comparison register. Use PWM mode 1 or mode 2 to set the OCxM domain center 110b or 111b, set the OCxPE bit to enable the preload register, and finally set the ARPE bit to enable the automatic reload of the preload register. When an update event occurs, the value of the preloaded register can be sent to the shadow register, so before the core counter starts counting, it is necessary to set the UG bit to initialize all the registers.

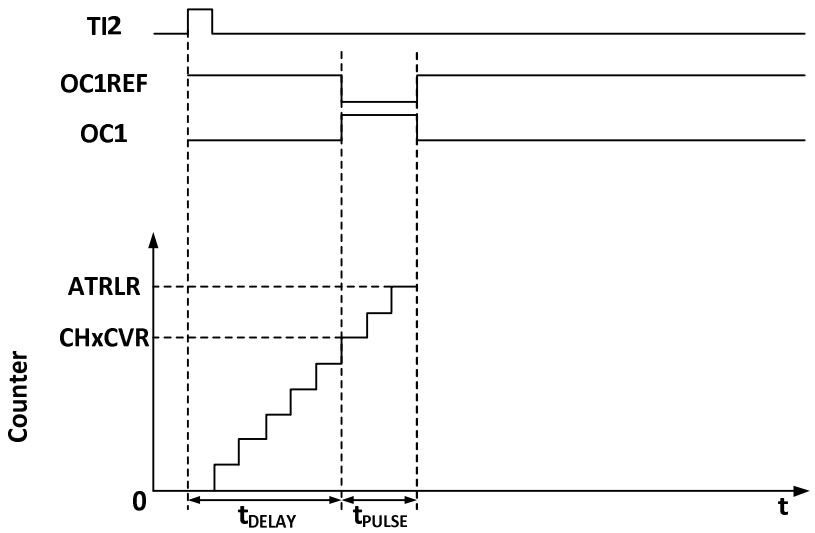
- Edge alignment

When edge alignment is used, the core counter is incremented. In the case of PWM mode 1, when the value of the core counter is greater than the comparison capture register, OCxREF rises to high. When the value of the core counter is less than the comparison capture register (for example, when the core counter increases to the value of R16_TIM3_ATRLR and returns to all zeros), OCxREF drops to low.

### 11.3.6 Monopulse Mode

Monopulse mode can respond to a specific event and generate a pulse after a delay. The delay and pulse width are programmable. Setting OPM bit can make the core counter stop when the next update event UEV is generated (the counter turns to 0).

Figure 11-5 Event generation and impulse response




As shown in Figure 11-5, it is necessary to detect the beginning of a rising edge on the input pin of TI2, and then generate a positive pulse with the length of Tpulse on OC1 after a delay of Tdelay:

1) Set TI2 to trigger. Set the CC2S domain to 01b, and map TI2FP2 to TI2； ; Set CC2P bit to 0b, and set TI2FP2 as rising edge detection; Set TS domain as 110b and TI2FP2 as trigger source; Set the SMS domain to 110b, and TI2FP2 is used to start the counter;

2) Tdelay is defined by the comparison capture register, and Tpulse is determined by the value of the automatic reload value register and the value of the comparison capture register.

### 11.3.7 Timer Synchronization Mode

Timer can output clock pulse (TRGO) and also receive input from other timers (ITRx). The source of ITRx of different timers (TRGO of other timers) is different. The internal trigger connection of timer is shown in Table 11-1.

Table 11-1 GTPM internal trigger connection

| Slave timer | ITR0 (TS=000) | ITR1 (TS=001) | ITR2 (TS=010) | ITR3 (TS=011) |
|---|---|---|---|---|
| TIM3 | TIM1 | TIM2 | USB | |

### 11.3.8 Dual Edge Capture Mode

The pulse can be measured by turning on the double-edge capture of the corresponding channel through the CAP_ED bit in the TIM3_CCER register.

For example, the pulse width is captured through channel 2, and the source of IC2 signal is selected (CC2S bit in TIM3_CHCTLR1 register is 11b) to enable the double-edge capture function of channel 2 (CAP_ED bit in TIM3_CCER register is 1). So far, the configuration of double edge capture has been completed.

The high and low level width values of the captured pulse can be read through the CH2CVR bit in the TIM3_CH2CVR register, and the level corresponding to the captured value can be indicated through the bit[16] in the TIM3_CH2CVR register by turning on the CAPLVL bit.

### 11.3.9 High Frequency Clock Counting Mode

Channels 1 and 2 of Timer 3 and the corresponding complementary channels support the use of the PLL clock as the core counter operating clock to generate PWM (the OCxM bit can only select PWM Mode 1 or PWM Mode 2), and only incremental counting mode is supported in this function; TIM3_SMCFGR register DT2[3:0] can be configured with the value of the dead time at the falling edge of OCXREF, and DT1[3:0 ] can be configured with the value of the dead time at the rising edge of OCXREF; the DT_DIV bit turns on the 4-division function of the dead time count clock to extend the dead time.

### 11.3.10 Debug Mode

When the system enters the debug mode, the timer can be controlled to continue running or stop according to the setting of DBG module.

## 11.4 Register Description

Table 11-2 TIM3-related registers list

| Name | Offset address | Description | Reset value |
|---|---|---|---|
| R16_TIM3_CTLR1 | 0x40000400 | TIM3 control register 1 | 0x0000 |
| R16_TIM3_SMCFGR | 0x40000408 | TIM3 slave mode control register | 0x0000 |
| R16_TIM3_DMAINTENR | 0x4000040C | TIM3 DMA/interrupt enable register | 0x0000 |
| R16_TIM3_INTFR | 0x40000410 | TIM3 interrupt status register | 0x0000 |
| R16_TIM3_SWEVGR | 0x40000414 | TIM3 event generation register | 0x0000 |

| | | | |
|---|---|---|---|
| R16_TIM3_CHCTLR1 | 0x40000418 | TIM3 compare/capture control register 1 | 0x0000 |
| R16_TIM3_CCER | 0x40000420 | TIM3 compare/capture enable register | 0x0000 |
| R16_TIM3_CNT | 0x40000424 | TIM3 counter | 0x0000 |
| R16_TIM3_PSC | 0x40000428 | TIM3 counting clock prescaler | 0x0000 |
| R16_TIM3_ATRLR | 0x4000042C | TIM3 auto-reload value register | 0xFFFF |
| R32_TIM3_CH1CVR | 0x40000434 | TIM3 compare/capture register 1 | 0x00000000 |
| R32_TIM3_CH2CVR | 0x40000438 | TIM3 compare/capture register 2 | 0x00000000 |

### 11.4.1 Control Register 1 (TIM3_CTLR1)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | CAPLVL | RW | In double-edge capture mode, the capture level indication is enabled.<br>1: Enable the indication function;<br>0: Disable the indication function.<br>*Note: When enabled, [16] of CHxCVR indicates the level corresponding to the capture value.* | 0 |
| 14 | CAPOV | RW | Capture value mode configuration.<br>1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture;<br>0: The capture value is the actual counter value. | 0 |
| [13:12] | Reserved | RO | Reserved | 0 |
| 11 | CO2NE | RW | Complementary output function with dead time-channel 2 and complementary channel output enable bits:<br>1: Complementary channel output and dead zone function of channel 2 are enabled;<br>0: The complementary channel output and dead-time function of channel 2 are disabled. | 0 |
| 10 | CO1NE | RW | Complementary output function with dead time-channel 1 and complementary channel output enable bits:<br>1: Complementary channel output and dead zone function of channel 1 are enabled;<br>0: The complementary channel output and dead-time function of channel 1 are disabled. | 0 |
| [9:8] | CKD[1:0] | RW | These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the sampling clock used by the digital filter.<br>00: Tdts=Tck_int;<br>01: Tdts= 2xTck_int;<br>10: Tdts= 4xTck_int;<br>11：Reserved. | 0 |
| 7 | ARPE | RW | Auto-reload preload enable bit.<br>1: Enable the Auto-reload value register (ATRLR).<br>0: disabled the Auto-reload value register (ATRLR). | 0 |
| [6:4] | Reserved | RO | Reserved | 0 |
| 3 | OPM | RW | Single pulse mode.<br>1: The counter stops when the next update event (Clearing the CEN bit) occurs.<br>0: The counter does not stop when the next update event occurs. | 0 |
| 2 | URS | RW | Update request source, by which the software selects the source of the UEV event.<br>1: If an update interrupt or DMA request is enabled, | 0 |

| | | | | |
|---|---|---|---|---|
| | | | only an update interrupt or DMA request is generated if the counter overflows/underflows.<br>0: If an update interrupt or DMA request is enabled, the update interrupt or DMA request is generated by any of the following events.<br>-Counter overflow/underflow<br>-Set the UG position<br>-Update generated from the mode controller | |
| 1 | UDIS | RW | Disable updates, the software allows/disables the generation of UEV events via this bit.<br>1: UEV is disabled. no update event is generated and the registers (ATRLR, PSC, CHCTLRx) maintain their values. If the UG bitis set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized.<br>0: UEV is allowed. update (UEV) events are generated by any of the following events:<br>- Counter overflow/underflow<br>-Set the UG position<br>-Update generated from the mode controller registers with caches are loaded with their preloaded values. | 0 |
| 0 | CEN | RW | Counter enable:<br>1: Enable the counter.<br>0: Disable the counter.<br>*Note: The external clock, gated mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.* | 0 |

### 11.4.2 Slave Mode Control Register (TIM3_SMCFGR)

Offset address: 0x08

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | DT2[3:0] | RW | Channel 2 dead time setting: T is the TIM3 module clock.<br>0000: Dead time is 1*T;<br>0001: Dead time is 2*T;<br>…<br>1110: Dead time is 15*T;<br>1111: Dead time is 16*T.<br>*Note: The dead time width must be less than the effective level width.* | 0 |
| [11:8] | DT1[3:0] | RW | Channel 1 dead time setting: T is the TIM3 module clock.<br>0000: Dead time is 1*T;<br>0001: Dead time is 2*T;<br>…<br>1110: Dead time is 15*T;<br>1111: Dead time is 16*T.<br>*Note: The dead time width must be less than the effective level width.* | 0 |
| 7 | Reserved | RO | Reserved | 0 |
| [6:4] | TS[2:0] | RW | Trigger select field, these 3 bits select the trigger input source used to synchronize the counter.<br>000: Internal trigger 0 (ITR0).<br>001: Internal trigger 1 (ITR1).<br>010: Internal trigger 2 (ITR2).<br>011: Internal trigger 3 (ITR3). | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 100: Edge detector of TI1 (TI1F_ED).<br>101: Filtered timer input 1 (TI1FP1).<br>110: Filtered timer input 2 (TI2FP2).<br>111: Reserved.<br>The above only changes when SMS is 0. | |
| 3 | Reserved | RO | Reserved. | 0 |
| [2:0] | SMS[2:0] | RW | Input mode selection field. Selects the clock and trigger mode of the core counter.<br>000: Driven by the internal clock CK_INT.<br>001: Reserved.<br>010: Reserved.<br>011: Reserved.<br>100: Reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers.<br>101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled.<br>110: Trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled.<br>111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter. | 0 |

### 11.4.3 DMA/Interrupt Enable Register (TIM2_DMAINTENR)

Offset address: 0x0C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | Reserved | RO | Reserved | 0 |
| 14 | TDE | RW | Trigger DMA request Enable bits:<br>1: Allow DMA request to be triggered;<br>0: Disable DMA request to be triggered. | 0 |
| [13:] | Reserved | RO | Reserved | 0 |
| 11 | CCDS | RW | 1: DMA request for CHxCVR is sent when an update event occurs;<br>0: Generate a DMA request for CHxCVR when a CHxCVR occurs. | 0 |
| 10 | CC2DE | RW | Compare/capture channel 2 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 2;<br>0: Disable the DMA request of compare capture channel 2. | 0 |
| 9 | CC1DE | RW | Compare/capture channel 1 DMA request enable bit.<br>1: Enable the DMA request of compare capture channel 1;<br>0: Disable the DMA request of compare capture channel 1. | 0 |
| 8 | UDE | RW | Updated DMA request enable bits:<br>1: Allow updated DMA requests;<br>0: Disable updated DMA requests. | 0 |
| 7 | Reserved | RO | Reserved | 0 |
| 6 | TIE | RW | Trigger interrupt enable bits:<br>1: Enable trigger interrupt;<br>0: Disable triggering interrupt. | 0 |
| [5:3] | Reserved | RO | Reserved | 0 |

| | | | | |
|---|---|---|---|---|
| 2 | CC2IE | RW | Compare capture channel 2 interrupt enable bit<br>1: Enable compare capture channel 2 interrupt;<br>0: Disable compare capture channel 2 interrupt. | 0 |
| 1 | CC1IE | RW | Compare capture channel 1 interrupt enable bit<br>1: Enable compare capture channel 1 interrupt;<br>0: Disable compare capture channel 1 interrupt. | 0 |
| 0 | UIE | RW | Update interrupt enable bit<br>1: Enable updated interrupt;<br>0: Disable updated interrupt. | 0 |

### 11.4.4 Interrupt Status Register (TIM3_INTFR)

Offset address: 0x10

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:11] | Reserved | RO | Reserved. | 0 |
| 10 | CC2OF | RW0 | Compare capture channel 2 over-capture flag bit | 0 |
| 9 | CC1OF | RW0 | Compare capture channel 1 over-capture flag bit is only used when the compare capture channel is configured for input capture mode. This flag is set by hardware and a software write of 0 clears this bit.<br>1: The state of CC1IF has been set when the counter value is captured to the capture compare register;<br>0: No over-capture generated. | 0 |
| [8:7] | Reserved | RO | Reserved. | 0 |
| 6 | TIF | RW0 | The trigger interrupt flag bit, which is set by hardware to this position when a trigger event occurs and cleared by software. Trigger events include the detection of a valid edge at the TRGI input when from a mode other than gated, or any edge in gated mode.<br>1: Trigger event generated;<br>0: No trigger event generated. | 0 |
| [5:3] | Reserved | RO | Reserved. | 0 |
| 2 | CC2IF | RW0 | Compare capture channel 2 interrupt flag bit. | 0 |
| 1 | CC1IF | RW0 | Compare capture channel 1 interrupt flag bit.<br>If the compare capture channel is configured in output mode, this bit is set by hardware when the counter value matches the compare value, except in centrosymmetric mode. This bit is cleared by software.<br>1: The value of the core counter matches the value of compare capture register 1; 0: no match occurs.<br>If compare capture channel 1 is configured in input mode, this bit is set by hardware when a capture event occurs and it is cleared by software or by reading the compare capture register.<br>1: The counter value has been captured compare capture register 1;<br>0: No input capture is generated. | 0 |
| 0 | UIF | RW0 | Update interrupt flag bit, this bit is set by hardware when an update event is generated and is cleared by software.<br>1: Update interrupt generated;<br>0: No update event is generated.<br>An update event will be generated in the following cases:<br>If UDIS=0, when the repeat counter value overflows or underflows; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | If URS=0, UDIS=0, when the UG bit is set, or when the counter core counter is reinitialized by software;<br>If URS=0, UDIS=0, when the counter CNT is reinitialized by a trigger event; | |

### 11.4.5 Event Generation Register (TIM3_SWEVGR)

Offset address: 0x14

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:7] | Reserved | RO | Reserved | 0 |
| 6 | TG | WO | The trigger event generation bit, which is set by software and cleared by hardware, is used to generate a trigger event.<br>1: Generate a trigger event, TIF is set, and the corresponding interrupts and DMAs are generated if enabled.<br>0: No action. | 0 |
| [5:3] | Reserved | RO | Reserved | 0 |
| 2 | CC2G | WO | Compare capture event generation bit 2. Generate compare capture event 2. | 0 |
| 1 | CC1G | WO | Compare capture event generation bit 1. Generate compare capture event 1. This bit is set by software and cleared by hardware.<br>It is used to generate a compare capture event.<br>1: Generate a compare capture event on compare capture channel 1.<br>If compare capture channel 1 is configured as output: set the CC1IF bit. Generate the corresponding interrupts and DMAs if they are enabled.<br>If compare capture channel 1 is configured as input: the current core counter value is captured to compare capture register 1; set the CC1IF bit and generate the corresponding interrupts and DMAs if they are enabled. If CC1IF is already set, set the CC1OF bit.<br>0: No action. | 0 |
| 0 | UG | WO | Update event generation bit to generate an update event. This bit is set by software and is automatically cleared by hardware.<br>1: Initialize the counter and generate an update event.<br>0: No action.<br>*Note: The prescaler counter is also cleared to zero, but the prescaler factor remains unchanged. The core counter is cleared if in centrosymmetric mode or incremental counting mode; if in decremental counting mode, the core counter takes the value of the reload value register.* | 0 |

### 11.4.6 Compare/Capture Control Register 1 (TIM3_CHCTLR1)

The channel can be used in input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register serve different purposes in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC2CE | OC2M[2:0] | | | OC2PE | OC2FE | CC2S[1:0] | | OC1CE | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |

| IC2F[3:0] | IC2PSC[1:0] | | IC1F[3:0] | IC1PSC[1:0] | |
|---|---|---|---|---|---|

Offset address: 0x18

Compare mode (pin direction is output).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | Reserved | RO | Reserved | 0 |
| [14:12] | OC2M[2:0] | RW | Compare the Capture Channel 2 mode setting field.<br>The 3 bits define the action of the output reference signal OC2REF, which determines the values of OC2, OC2N. OC2REF is active high, while the active levels of OC2 and OC2N depend on the CC2P, CC2NP bits.<br>000: Freeze. Comparison of the value of the capture register with the value of the comparison between the core counters does not work for OC1REF.<br>001: Force to set to valid level. Forcing OC1REF high when the core counter has the same value as the comparison capture register 2.<br>010: Force to set to invalid level. Forcing OC1REF low when the value of the core counter is the same as the comparison capture register 2.<br>011: Flip. Flips the level of OC1REF when the core counter is the same as the value of compare capture register 2.<br>100: Forced to invalid level. Forces OC1REF to low.<br>101: Force to valid level. Force OC1REF to high.<br>110: PWM mode 1: When counting up, once the core counter is less than the value of the comparison capture register, channel 2 is an effective level, otherwise it is an invalid level; When counting down, once the core counter is greater than the value of the comparison capture register, channel 2 is at an invalid level (OC2REF=0), otherwise it is at an effective level (OC2REF = 1);<br>111: PWM mode 2: When counting up, once the core counter is less than the value of the comparison capture register, channel 2 is at an invalid level, otherwise it is at an effective level; When counting down, once the core counter is greater than the value of the comparison capture register, Channel 2 is active (OC2REF=1), otherwise it is inactive (OC2REF=0).<br>*Note: This bit cannot be modified once the LOCK level is set to 3 and CC1S=00b. In PWM mode 1 or PWM mode 2, the OC1REF level is changed only when the comparison result is changed or when switching from freeze mode to PWM mode in the output comparison mode.* | 0 |
| 11 | OC2PE | RW | Compare Capture Register 2 preload enable bit.<br>1: Turn on the preload function of the compare capture register 2, the read and write operations operate only on the preload register, and the preload value of the compare capture register 2 is loaded into the current shadow register when the update event comes.<br>0: Disable the preload function of compare capture register 2. The compare capture register 2 can be written at any time, and the newly written value takes effect immediately.<br>*Note: Once the LOCK level is set to 3 and CC2S=00,* | 0 |

| | | | | |
|---|---|---|---|---|
| | | | *this bit cannot be modified. PWM mode can be used only in single pulse mode (OPM=1) without confirming the pre-load register, otherwise its action is not determined.* | |
| 10 | OC2FE | RW | Compare Capture Channel 2 fast enable bit, this bit is used to speed up the response of the compare capture channel output to trigger input events.<br>1: The active edge of the input to the flipflop acts as if a comparison match has occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the valid edge of the sample trigger and the output of the compare capture channel 2 is reduced to 3 clock cycles.<br>0: Based on the value of the counter and compare capture register 1, compare capture channel 2 operates normally, even if the flip-flop is open. The minimum delay to activate the compare capture channel 2 output is 5 clock cycles when the input of the flipflop has a valid edge.<br>OC2FE only works when the channel is configured to PWM1 or PWM2 mode. | 0 |
| [9:8] | CC2S[1:0] | RW | Compare capture channel 2 input selection fields.<br>00: comparison capture channel 2 is configured as an output.<br>01: comparison capture channel 2 is configured as an input and IC2 is mapped on TI2.<br>10: comparison capture channel 2 is configured as an input and IC2 is mapped onTI1.<br>11: Compare Capture Channel 2 is configured as an input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (By the TS bit).<br>*Note: Compare capture channel 2 is writable only when the channel is off (When CC2E is zero).* | 0 |
| 7 | Reserved | RO | Reserved | 0 |
| [6:4] | OC1M[2:0] | RW | Compare capture channel 1 mode setting field. | 0 |
| 3 | OC1PE | RW | Compare capture register 1 preload enable bit. | 0 |
| 2 | OC1FE | RW | Compare capture channel 1 fast enable bit. | 0 |
| [1:0] | CC1S[1:0] | RW | Compare capture channel 1 input selection fields. | 0 |

Capture mode (pin direction is input).

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | IC2F[3:0] | RW | The input capture filter 2 configuration field, these bits set the sampling frequency of the TI1 input and the digital filter length. The digital filter consists of an event counter, which records N events and then generates a jump in the output.<br>0000: No filter, sampled at fDTS.<br>1000: Sampling frequency Fsampling = Fdts/8, N = 6.<br>0001: Sampling frequency Fsampling=Fck_int, N=2.<br>1001: Sampling frequency Fsampling = Fdts/8, N = 8.<br>0010: Sampling frequency Fsampling=Fck_int, N=4.<br>1010: Sampling frequency Fsampling = Fdts/16, N = 5.<br>0011: Sampling frequency Fsampling=f=Fck_int, N=8.<br>1011: Sampling frequency Fsampling = Fdts/16, N = 6.<br>0100: Sampling frequency Fsampling = Fdts/2, N = 6. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1100: Sampling frequency Fsampling = Fdts/16, N = 8.<br>0101: Sampling frequency Fsampling = Fdts/2, N = 8.<br>1101: Sampling frequency Fsampling = Fdts/32, N = 5.<br>0110: Sampling frequency Fsampling = Fdts/4, N = 6.<br>1110: Sampling frequency Fsampling = Fdts/32, N = 6.<br>0111: Sampling frequency Fsampling = Fdts/4, N = 8.<br>1111: Sampling frequency Fsampling=Fdts/32, N=8. | |
| [11:10] | IC2PSC[1:0] | RW | Compare capture channel 2 prescaler configuration field, these 2 bits define the prescaler coefficient for compare capture channel 2. Once CC1E = 0, the prescaler is reset.<br>00: without prescaler, one capture is triggered for each edge detected on the capture input.<br>01: Capture triggered every 2 events.<br>10: Capture triggered every 4 events.<br>11: Capture is triggered every 8 events. | 0 |
| [9:8] | CC2S[1:0] | RW | Compare the capture channel 2 input selection field, these 2 bits define the direction of the channel (Input/output), and the selection of the input pin.<br>00: Comparative capture channel 1 channel is configured as an output.<br>01: Comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI1.<br>10: Comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI2.<br>11: The compare capture channel 1 channel is configured as an input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (By the TS bit).<br>*Note: CC1S is writable only when the channel is off (CC1E is 0).* | 0 |
| [7:4] | IC1F[3:0] | RW | Input capture filter 1 configuration field. | 0 |
| [3:2] | IC1PSC[1:0] | RW | Compare capture channel 1 prescaler configuration field. | 0 |
| [1:0] | CC1S[1:0] | RW | Compare capture channel 1 input selection fields. | 0 |

### 11.4.7 Compare/Capture Enable Register (TIM3_CCER)

Offset address: 0x20

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:14] | Reserved | RO | Reserved | 0 |
| 13 | HSCK_MODE | RW | Counter clock selection bits. Use this function to enable the PLL clock:<br>1: The core counters of channels 1 and 2 use the PLL clock as the working clock;<br>0: Channels 1 and 2 maintain the original clock signal as the working clock. | 0 |
| 12 | DT_DIV | RW | Dead time counting clock frequency division enable:<br>1: Divide the clock by 4;<br>0: The clock is not divided. | 0 |
| 11 | ADC_CAP | RW | Capture ADC module signal edge function enabled:<br>1: Alternate channel 1 double edge capture function, and the counter is cleared after capture;<br>0: The function is turned off, and the channel 1 double edge capture function is normal. | 0 |
| 10 | CAP_ED | RW | Channel 2 double edge capture function enable:<br>1: Channel 2 double edge capture function enable;<br>0: Channel 2 double edge capture function disable. | 0 |

| | | | | |
|---|---|---|---|---|
| 9 | CO2NP | RW | Complementary output function-Channel 2 complementary channel output polarity setting bit:<br>1: Active low;<br>0: Active high. | 0 |
| 8 | CO1NP | RW | Complementary output function-Channel 1 complementary channel output polarity setting bit:<br>1: Active low;<br>0: Active high. | 0 |
| 7 | CO2P | RW | Complementary output function-Channel 2 output polarity setting bit:<br>1: Active low;<br>0: Active high. | 0 |
| 6 | CO1P | RW | Complementary output function-Channel 1 output polarity setting bit:<br>1: Active low;<br>0: Active high. | 0 |
| 5 | CC2P | RW | Compare capture channel 2 output polarity setting bit. | 0 |
| 4 | CC2E | RW | Compare capture channel 2 output enable bit. | 0 |
| [3:2] | Reserved | RO | Reserved | 0 |
| 1 | CC1P | RW | Compare capture channel 1 output polarity setting bit.<br>CC1 channel is configured as output:<br>1: OC1 is active low;<br>0: OC1 is active high.<br>CC1 channel is configured as input:<br>This bit selects whether IC1 or the inverted signal of IC1 is used as the trigger or capture signal.<br>1: Inversion: Capture occurs at the falling edge of IC1; When used as an external trigger, IC1 is inverted;<br>0: Non-inverting: capture occurs at the rising edge of IC1; When used as an external trigger, IC1 does not invert. | 0 |
| 0 | CC1E | RW | Compare capture channel 1 output enable bit.<br>CC1 channel is configured as output:<br>1: On. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE.<br>0: Off. OC1 prohibits output, so the output level of OC1 depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits.<br>CC1 channel is configured as input:<br>This bit determines whether the value of the counter can be captured in the TIM1_CH1CVR register.<br>1: Capture enabled;<br>0: Capture disabled. | 0 |

### 11.4.8 Streamlined timer counter (TIM3_CNT)

Offset address: 0x24

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CNT[15:0] | RW | The real-time value of the counter of the timer. | 0 |

### 11.4.9 Counting Clock Prescaler (TIM3_PSC)

Offset address: 0x28

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | PSC[15:0] | RW | The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input | 0 |

| | | | | |
|---|---|---|---|---|
| | | | frequency of the divider/(PSC+1). | |

### 11.4.10 Auto-reload Register (TIM3_ATRLR)

Offset address: 0x2C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | ATRLR[15:0] | RW | The value of ATRLR[15:0] will be loaded into the counter. Please read section 11.2.4 when ATRLR is activated and updated. When ATRLR is empty, the counter stops. | 0xFFFF |

### 11.4.11 Compare/Capture Register 1 (TIM3_CH1CVR)

Offset address: 0x34

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH1CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL1 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH1CVR[15:0] | RW | Compare/capture the value of register channel 1. | 0 |

### 11.4.12 Compare/Capture Register 2 (TIM3_CH2CVR)

Offset address: 0x38

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LEVEL2 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH2CVR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | LEVEL2 | RO | Level indication bit corresponding to the captured value | 0 |
| [15:0] | CH2CVR[15:0] | RW | Compare/capture the value of register channel 2. | 0 |

# Chapter 12 Universal Asynchronous Receiver Transmitter (USART)

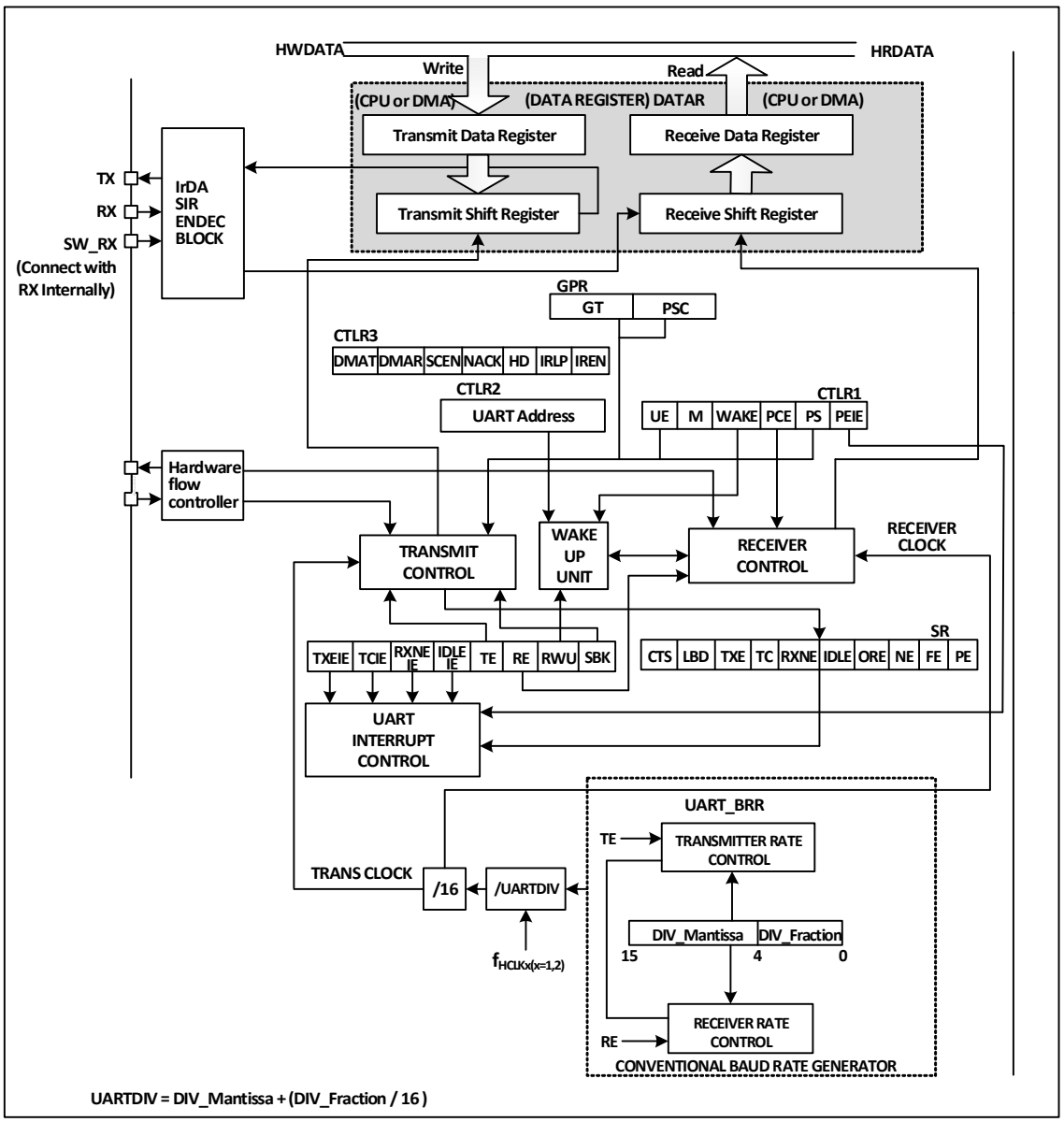
The module contains 1 Universal Asynchronous Transceiver (USART).

## 12.1 Main Features

- Full-duplex or half-duplex synchronous or asynchronous communication
- NRZ data format
- Fractional baud rate generator, up to 4.5Mbps
- Programmable data length
- Configurable stop bits
- Support LIN, IrDA encoders
- Support DMA
- Multiple interrupt sources

## 12.2 Overview

Figure 12-1 Block diagram of a general-purpose synchronous/asynchronous transceiver




When TE (transmit enable bit) is set, the data in the transmit shift register is output on the TX pin. When sending, least significant bit is the first to be moved out, and each data frame starts with a low-level start bit, then the transmitter sends an eight-or nine-bit data word according to the setting on the m (word length) bit, and finally a configurable number of stop bits. If parity bits are provided, the last bit of the data word is the parity bit. After TE is set, an idle frame is sent, which is 10-bit or 11-bit high level and contains stop bits. The disconnect frame is 10-bit or 11-bit low, followed by a stop bit.

## 12.3 Baud Rate Generator

The baud rate of the transceiver = HCLK/(16*USARTDIV), HCLK is the clock of HB. The value of USARTDIV is determined by the 2 fields DIV_M and DIV_F in USART_BRR, which is calculated by the formula. The formula is as follows.

USARTDIV = DIV_M+(DIV_F/16)

It should be noted that the baud rate generated by the baud rate generator may not necessarily generate exactly what the user needs, and there may be deviations in this. In addition to trying to get close values, the method of reducing the deviation can also be to increase the HB clock. For example, when setting the baud rate to 115200bps, the value of UARTDIV is set to 39.0625. At the highest frequency (72MHz), you can get a baud rate of exactly 115200bps. However, if you need a baud rate of 921600bps, the calculated UARTDIV is 4.88, but in fact, the closest value filled in UART_BRR is only 4.875. The actual baud rate generated is 923076bps, and the error reaches 0.16%.

When the serial port waveform sent by the sender is transmitted to the receiver, there is a certain error in the baud rate between the receiver and the sender. The error mainly comes from three aspects: the actual baud rate of the receiver and the sender is inconsistent; the clock of the receiver and the sender is incorrect; changes in the waveform line. The receiver of the peripheral module has a certain reception tolerance capability. When the sum of the total deviations generated in the above three aspects is less than the module's tolerance capability limit, this total deviation does not affect transmission and reception. The tolerance limit of the module is affected by whether the fractional baud rate and M bits (data domain word length) are used. Using fractional baud rate and using 9-bit data domain length will reduce the tolerance limit, but not less than 3%.

## 12.4 1-Wire Half-Duplex Mode

Half-duplex mode supports the use of a single pin (TX pin only) for receive and transmit, with the TX and RX pins connected internally on the chip.

The way to turn on the half-duplex mode is to set the HDSEL bit of the control register 3 (R32_UART1_CTLR3), but at the same time, the LIN mode and infrared mode need to be turned off, that is, to ensure that the IREN bit is in the reset state, which is in the control register 3 (R32_UART1_CTLR3).

After setting it to half-duplex mode, the IO port of TX needs to be set to alternate output high mode. When TE is asserted, as long as the data is written to the data register, it will be sent out. It is particularly important to note that half-duplex mode may cause bus conflicts when multiple devices use single bus transmission and reception, which requires users to avoid using software.

## 12.5 IrDA

UART module supports controlling IrDA infrared transceiver for physical layer communication. To use IrDA, the LINEN, STOP and HDSEL bits must be cleared. NRZ (non-return-to-zero) coding is used between UART module and SIR physical layer (infrared transceiver), which supports up to 115200bps.

IrDA is a half-duplex protocol. If UART is sending data to SIR physical layer, IrDA decoder will ignore the new infrared signal. If UART is receiving data from SIR, SIR will not accept the signal from UART. The level logic sent by UART to SIR is different from that sent by SIR to UART. In SIR receiving logic, the high level is 1 and the low level is 0, but in SIR sending logic, the high level is 0 and the low level is 1.

## 12.6 DMA

UART module supports DMA function, which can be used to realize fast and continuous transmission and reception.

When DMA is enabled and TXE is set, DMA will write data to the send buffer from the set memory space. When using DMA reception, every time RXNE is set, DMA will transfer the data in the reception buffer to a specific memory space.

## 12.7 Interrupts

The UART module supports a variety of interrupt sources, including transmit data register empty (TXE), CTS, transmit complete (TC), receive data ready (RXNE), data overflow (ORE), line idle (IDLE), parity error (PE), disconnect flag (LBD), noise (NE), overflow for multi-buffered communication (ORE), and frame error (FE), among others.

Table 12-1 Relationship between interrupts and corresponding enable bits

| Interrupt source | Enable bit |
|---|---|
| Transmit data register empty (TXE) | TXEIE |
| CTS interrupt (CTS) | CTSIE |
| Transmission complete (TC) | TCIE |
| Ready to receive data (RXNE) | RXNEIE |
| Overrun error detected (ORE) | |
| Idle line detected (IDLE) | IDLEIE |
| Parity error (PE) | PEIE |
| LIN break (LBD) | LBDIE |
| Noise flag（NE） | EIE |
| Overflow of multi-buffered communication (ORE) | |
| Frame error (FE) for multibuffered communication | |

## 12.8 Register Description

Table 12-2 UART-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_UART1_STATR | 0x40013800 | UART status register | 0x000000C0 |
| R32_UART1_DATAR | 0x40013804 | UART data register | 0x00000000 |
| R32_UART1_BRR | 0x40013808 | UART baud rate register | 0x00000000 |
| R32_UART1_CTLR1 | 0x4001380C | UART control register 1 | 0x00000000 |
| R32_UART1_CTLR2 | 0x40013810 | UART control register 2 | 0x00000000 |
| R32_UART1_CTLR3 | 0x40013814 | UART control register 3 | 0x00000000 |
| R32_UART1_GPR | 0x40013818 | UART guard time and prescaler register | 0x00000000 |

### 12.8.1 UART Status Register (UART1_STATR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | RX_BUSY | CTS | LBD | TXE | TC | RXNE | IDLE | ORE | NE | FE | PE |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:11] | Reserved | RO | Reserved | 0 |
| 10 | RX_BUSY | RO | Receiving state indication bit:<br>1: This bit when operating in the receive state;<br>0: In non-reception state. | 0 |
| 9 | CTS | RW0 | CTS state change flag. If the CTSE bit is set, this bit will be set high by hardware when the nCTS output state changes. It is cleared to zero by software. If the CTSIE bit is already set, an interrupt will be generated.<br>1: There is a change on the nCTS status line;<br>0: There is no change on the nCTS status line. | 0 |
| 8 | LBD | RW0 | LIN Break detection flag. This bit is set by hardware when LIN Break is detected. Cleared by software. If LBDIE has been set, an interrupt will be generated.<br>1: LIN Break is detected;<br>0: LIN Break is not detected. | 0 |
| 7 | TXE | RO | Transmit data register empty flag. This bit is set by hardware when the data in the TDR register is transferred to the shift register by hardware. If TXEIE has been set, an interrupt will be generated and the data register will be written and this bit will be reset.<br>1: The data has been transferred to the shift register;<br>0: The data has not been transferred to the shift register. | 1 |
| 6 | TC | RW0 | Transmit completion flag. When a frame containing data is transmitted and TXE is set, the hardware will set this bit. If TCIE is set, a corresponding interrupt will be generated. The software will clear this bit after reading this bit and writing to the data register. You can also directly write 0 to clear this bit.<br>1: The transmitting is completed;<br>0: The transmitting is not yet completed. | 1 |
| 5 | RXNE | RW0 | Read the data register non-empty flag. When the data in the shift register is transferred to the data register, this bit will be set by hardware. If RXNEIE has been set, the corresponding interrupt will also be generated. Reading the data register clears this bit. You can also write 0 directly to clear this bit.<br>1: The data has been received and can be read;<br>0: The data has not been received yet. | 0 |
| 4 | IDLE | RO | Bus idle flag. This bit will be set by hardware when the bus is free. If IDLEIE has been set, the corresponding interrupt will be generated. Reading the status register and then the data register clears this bit.<br>1: Bus is idle;<br>0: Bus idle is not detected.<br>*Note: This bit will not be set again until RXNE is set.* | 0 |
| 3 | ORE | RO | Overload error flag. This bit will be set when there is data in the receive shift register that needs to be transferred to the data register, but there is still data in the receive field of the data register that has not been read out. If RXNEIE is set, the corresponding interrupt will also be generated.<br>1: An overload error occurs; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0: There is no overload error.<br>*Note: In case of an overload error, the value of the data register is not lost, but the value of the shift register is overwritten. If the EIE able bit is set, the ORE flag position bit generates an interrupt in multi-buffer communication mode.* | |
| 2 | NE | RO | Noise error flag. It is set by hardware when the noise error flag is detected. The operation of reading the status register and then reading the data register resets this bit.<br>1: Noise detected.<br>0: No noise is detected.<br>*Note: This bit does not generate an interrupt. If the EIE bit is set, the FE flag position bit generates an interrupt in multi-buffer communication mode.* | 0 |
| 1 | FE | RO | Frame error flag. This bit will be set by hardware when a synchronization error, excessive noise or disconnect character is detected. Reading this bit and then reading the data register operation will reset this bit.<br>1: Frame error detected.<br>0: No frame error detected.<br>*Note: This bit will not generate an interrupt. If the EIE bit is set, the FE flag position bit will generate an interrupt in multi-buffer communication mode.* | 0 |
| 0 | PE | RO | Checksum error flag. In receive mode, hardware sets this bit if a parity check error is generated. A read of this bit and then a read of the data register operation resets this bit. Before clearing this bit, software must wait for the RXNE flag bit to be set. If the PEIE has been set previously, then this bit being set generates a corresponding interrupt.<br>1: A parity error.<br>0: No inspection error. | 0 |

### 12.8.2 UART Data Register (UART1_DATAR)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | DR[8:0] | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:9] | Reserved | RO | Reserved | 0 |
| [8:0] | DR[8:0] | RW | Data register. This register is actually the receive data register (RDR) and send register (TDR) two registers composed of DR read and write operation start is read receive register (RDR) and write send register (TDR) respectively. | X |

### 12.8.3 UART Baud Rate Register (UART1_BRR)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV_Mantissa[11:0] | | | | | | | | | | | | DIV_Fraction[3:0] | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0 |
| [15:4] | DIV_Mantissa[11:0] | RW | These 12 bits define the integer part of the dividing factor of the frequency divider. | 0 |
| [3:0] | DIV_Fraction[3:0] | RW | These 4 bits define the fractional part of the dividing factor of the frequency divider. | 0 |

### 12.8.4 UART Control Register 1 (UART1_CTLR1)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | UE | M | WAKE | PCE | PS | PEIE | TXEIE | TCIE | RXNEIE | IDLEIE | TE | RE | RWU | SBK |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:14] | Reserved | RO | Reserved | 0 |
| 13 | UE | RW | UART enable bit. When this bit is cleared, both the UART divider and the output stop working after the current byte transfer is completed. | 0 |
| 12 | M | RW | Word long bit.<br>1: 9 data bits; 0: 8 data bits. | 0 |
| 11 | WAKE | RW | Wake-up bit. This bit determines the method of waking up the UART.<br>1: Address marker; 0: Bus idle. | 0 |
| 10 | PCE | RW | The parity bit is enabled. For the receiver, it is the parity check of the data; for the sender, it is the insertion of the parity bit. Once this bit is set, the parity bit enable will take effect only after the current byte transmission is completed. | 0 |
| 9 | PS | RW | Parity selection. 0 means even parity, 1 means odd parity. When this bit is set, the parity bit enable will take effect only after the current byte transmission is completed. | 0 |
| 8 | PEIE | RW | Parity check interrupt enable bit. This bit indicates that parity check error interrupts are allowed. | 0 |
| 7 | TXEIE | RW | TXE interrupt enable. This bit indicates that a TXE interrupt is allowed to be generated. | 0 |
| 6 | TCIE | RW | Transmit completion interrupt enable. This bit indicates that the transmit completion interrupt is allowed to be generated. | 0 |
| 5 | RXNEIE | RW | RXNE interrupt enable. This bit indicates that a RXNE interrupt is allowed to be generated. | 0 |

| | | | | |
|---|---|---|---|---|
| 4 | IDLEIE | RW | IDLE interrupt enable. This bit allows IDLE interrupt to be generated. | 0 |
| 3 | TE | RW | Transmit enable. Setting this bit will enable the transmitter. | 0 |
| 2 | RE | RW | Receive enable. Setting this bit enables the receiver, which starts detecting the start bit on the RX pin. | 0 |
| 1 | RWU | RW | Receiver wakeup. This bit determines whether to place the UART in silent mode.<br>1: The receiver is in silent mode.<br>0: The receiver is in normal operation mode.<br>*Note 1: Before setting the RWU bit, the UART needs to receive a data byte first, otherwise it cannot be woken up by bus idle in silent mode.*<br>*Note 2: When configured as address mark wake-up, the RWU bit cannot be modified by software when RXNE is set.* | 0 |
| 0 | SBK | RW | Send break bit.<br>Set this bit to send break character. It is reset by hardware on the stop bit of the break frame.<br>1: Send; 0: Do not send. | 0 |

### 12.8.5 UART Control Register 2 (UART1_CTLR2)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | LINEN | STOP[1:0] | | Reserved | | | | | LBDIE | LBDL | Reserved | ADD[3:0] | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:15] | Reserved | RO | Reserved | 0 |
| 14 | LINEN | RW | The LIN mode enable bit, set to enable LIN mode. In LIN mode, the SBK bit can be used to send LIN sync break symbols and to detect LIN sync break characters. | 0 |
| [13:12] | STOP[1:0] | RW | STOP bits.<br>These 2 bits are used for programming the stop bits.<br>00: 1 Stop bit<br>01: 0.5 Stop bit<br>10: 2 Stop bits<br>11: 1.5 Stop bit | 0 |
| [11:7] | Reserved | RO | Reserved | 0 |
| 6 | LBDIE | RW | LIN Break detection interrupt enable, this position bit enables interrupts caused by LBD. | 0 |
| 5 | LBDL | RW | LIN Break detection length, this bit is used to select whether the disconnect detection is 11 bits or 10 bits.<br>1: 11-bit disconnector detection.<br>0: 10-bit disconnector detection. | 0 |
| 4 | Reserved | RW | Reserved | 0 |
| [3:0] | ADD[3:0] | RW | Address field to set the UART node address of this | 0 |

| | | | | |
|---|---|---|---|---|
| | | | device. Used in silent mode under multi-processor communication to wake up a USART device using the address token. | |

### 12.8.6 UART Control Register 3 (UART1_CTLR3)

Offset address: 0x14

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | CTSIE | CTSE | RTSE | DMAT | DMAR | Reserved | | HDSEL | IRLP | IREN | EIE |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:11] | Reserved | RO | Reserved. | 0 |
| 10 | CTSIE | RW | CTSIE interrupt enable bit, when set this bit will generate an interrupt when CTS is set. | 0 |
| 9 | CTSE | RW | CTS enable bit, setting this bit will enable CTS flow control. | 0 |
| 8 | RTSE | RW | RTS enable bit, setting this bit will enable RTS flow control. | 0 |
| 7 | DMAT | RW | DMA transmit enable bit. This position 1 uses DMA when transmitting. | 0 |
| 6 | DMAR | RW | DMA receive enable bit. This position 1 uses DMA when receiving. | 0 |
| [5:4] | Reserved | RO | Reserved. | 0 |
| 3 | HDSEL | RW | Half duplex mode select bit, set this bit to select half duplex mode. | 0 |
| 2 | IRLP | RW | Infrared low-power select bit, set this bit to enable low-power mode when infrared is selected. | 0 |
| 1 | IREN | RW | Infrared enable bit, set this bit to enable infrared mode. | 0 |
| 0 | EIE | RW | Error enable interrupt bit, setting this bit generates an interrupt if FE, ORE or NE is set provided that DMAR is set. | 0 |

### 12.8.7 UART Guard Time and Prescaler Register (UART1_GPR)

Offset address: 0x18

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | PSC[7:0] | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:8] | Reserved | RO | Reserved | 0 |
| [7:0] | PSC[7:0] | RW | Prescaler value field.<br>In IR low-power mode, the source clock is divided by this value (All 8 bits valid) and a value of 0 indicates a hold;<br>In IR normal mode, this bit can only be set to 1; | 0 |

# Chapter 13 Inter-integrated Circuit (I2C) interface

The internal integrated circuit bus (I2C or IIC) is widely used for communication between microcontrollers and sensors and other off-chip modules. It inherently supports multi-master and multi-slave modes and can communicate at 100KHz (Standard) and 400KHz (Fast) using just two wires (SDA and SCL).

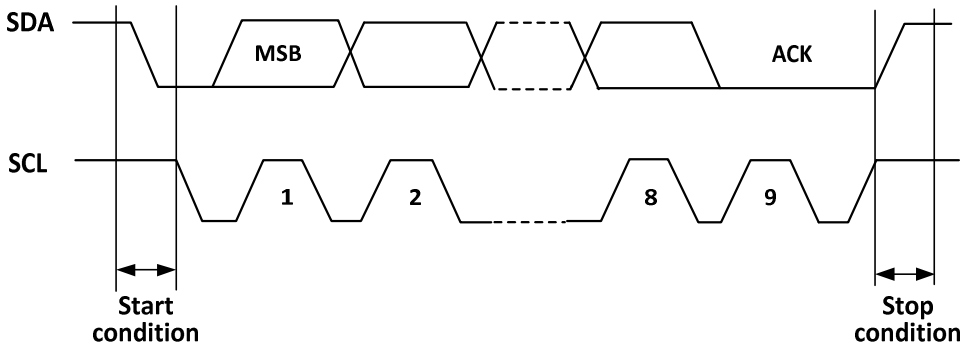
## 13.1 Main Features

- Support master and slave modes
- Support 7-bit or 10-bit addresses
- Slave devices support dual 7-bit addresses
- Support 2 speed modes: 100kHz and 400kHz
- Multiple status modes, multiple error flags
- Support extended clock function
- 2 interrupt vectors
- Support DMA
- Support PEC.

## 13.2 Overview

I2C is a half-duplex bus that can only operate in one of the following 4 modes at the same time: master device transmit mode, master device receive mode, slave device transmit mode and slave device receive mode. the I2C module works in slave mode by default and automatically switches to master mode when a start condition is generated and to slave mode when arbitration is lost or a stop signal is generated. the I2C module supports multi-master functionality. When working in master mode, the I2C module actively emits data and addresses. Both data and address are transmitted in 8-bit units, with the high bit before and the low bit after. After the start event is a one-byte (In 7-bit address mode) or two-byte (In 10-bit address mode) address, and for every 8-bit data or address sent by the host, the slave needs to reply with an answer ACK, which pulls the SDA bus low, as shown in Figure 13-1.

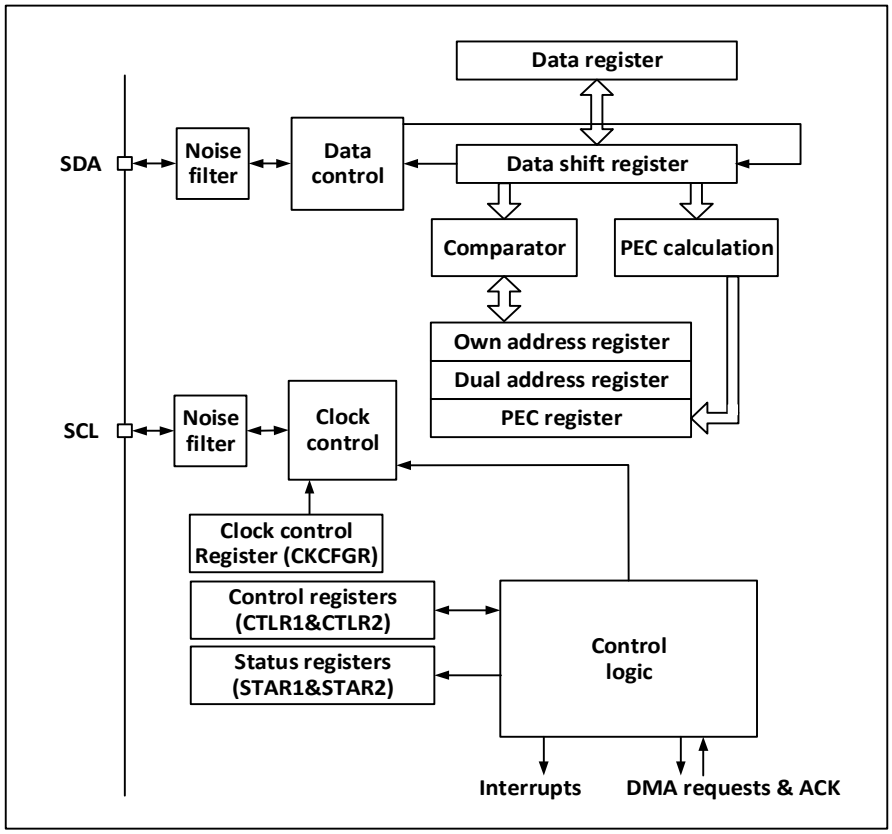
Figure 13-1 I2C Timing Diagram




For normal use, the correct clock must be input to I2C. In standard mode, the minimum input clock is 2MHz, and in fast mode, the minimum input clock is 4MHz.

Figure 13-2 is the functional block diagram of I2C module.

Figure 13-2 I2C Functional block Diagram




## 13.3 Master Mode

In master mode, the I2C module dominates the data transfer and outputs the clock signal, and the data transfer starts with a start event and ends with an end event. The steps to use master mode communication are:

1) Setting the correct clock in control register 2 (R16_I2C1_CTLR2) and clock control register (R16_I2C1_CKCFGR).
2) Set an appropriate rising edge in the rising edge register (R16_I2C1_RTR);
3) Set the PE bit in the control register (R16_I2C1_CTLR1) to start the peripheral;
4) Set the START bit in the control register (R16_I2C1_CTLR1) to generate the start event.

After the START bit is set, the I2C module will automatically switch to the main mode, the MSL bit will be set, and the start event will be generated. After the start event is generated, the SB bit will be set, and if the ITEVTEN bit (in R16_I2C1_CTLR2) is set, an interrupt will be generated. At this time, the status register 1(R16_I2C1_STAR1) should be read, and the SB bit will be cleared automatically after writing from the address to the data register;

If the 10-bit address mode is used, the write data register sends a header sequence (the header sequence is 11110xx0b, where xx bits are the most significant two bits of the 10-bit address).

After sending the header sequence, the ADD10 bit of the status register will be set. If the ITEVTEN bit has been set, an interrupt will be generated. At this time, read the R16_I2C1_STAR1 register, write the second address byte to the data register, and then clear the ADD10 bit.

Then write the data register and send the second address byte. After the second address byte is sent, the ADDR bit of the status register will be set. If the ITEVTEN bit has been set, an interrupt will be generated. At this time, read

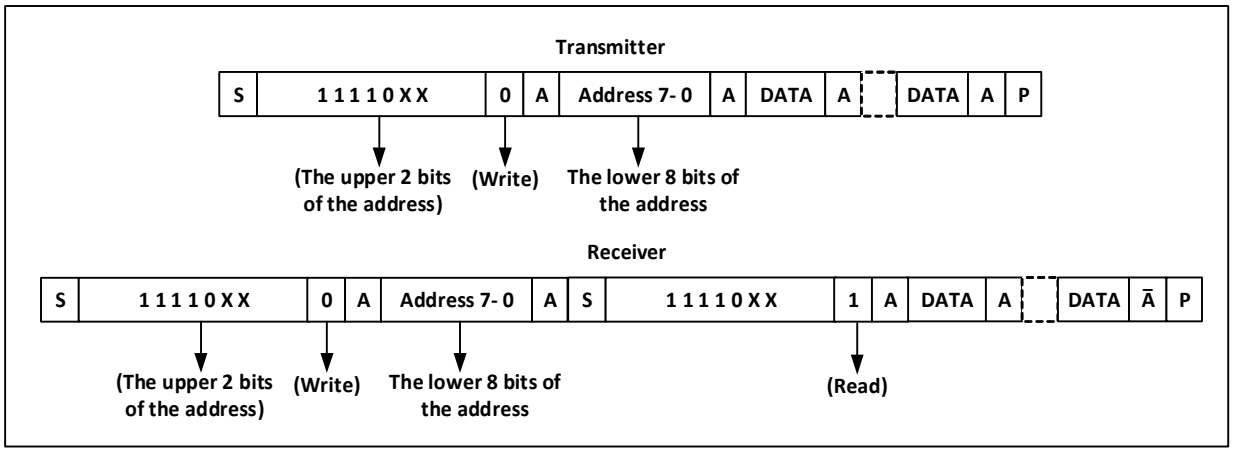
the R16_I2C1_STAR1 register and then read the R16_I2C1_STAR2 register again to clear the ADDR bit.

If the 7-bit address mode is used, the write data register sends address bytes, and after the address bytes are sent, the ADDR bit of the status register will be set. If the ITEVTEN bit has been set, an interrupt will be generated. At this time, the R16_I2C1_STAR1 register should be read and then the R16_I2C1_STAR2 register should be read again to clear the ADDR bit.

In the 7-bit address mode, the first byte sent is the address byte, the first 7 bits represent the address of the target slave device, and the eighth bit determines the direction of subsequent messages. 0 means that the master device writes data to the slave device, and 1 means that the master device reads information from the slave device.

In 10-bit address mode, as shown in Figure 13-3, in the send address phase, the first byte is 11110xx0, xx is the highest 2 bits of the 10-bit address, and the second byte is the lower 8 bits of the 10-bit address. If subsequently enter the master device transmit mode, continue to send data; if subsequently ready to enter the master device receive mode, you need to re-send a start condition, follow to send a byte as 11110xx1, and then enter the master device receive mode.

Figure 13-3 Schematic diagram of master transmitting and receiving data at 10-bit address

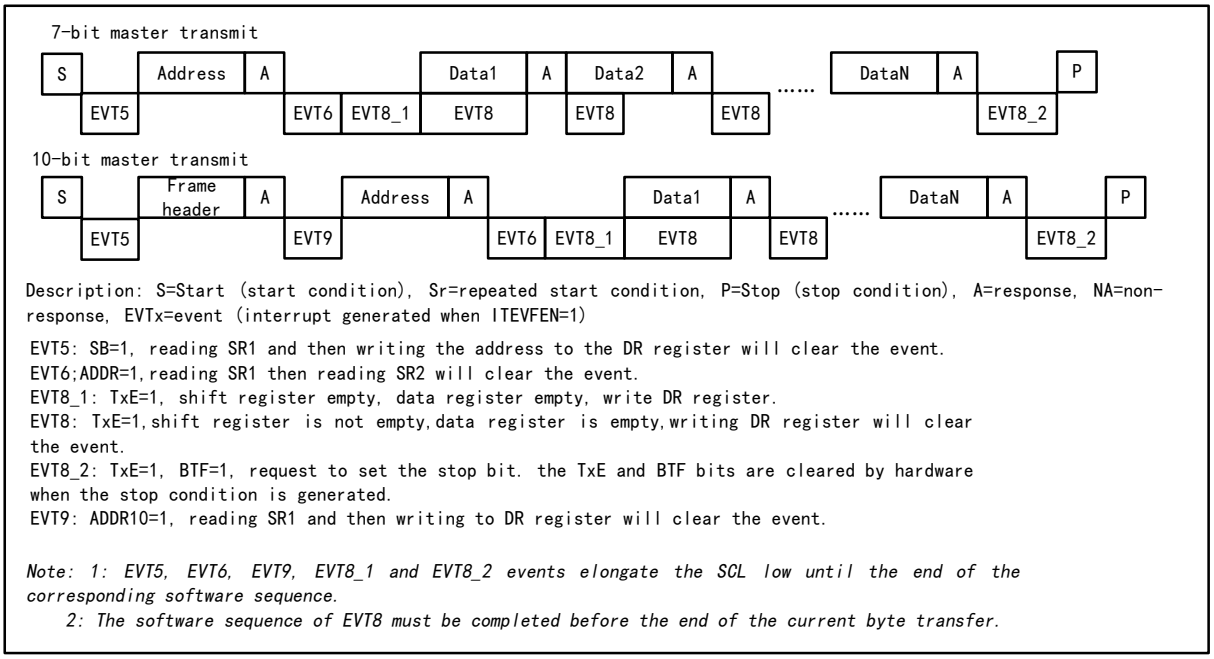



**Master transmit mode:**

The master device's internal shift register sends data from the data register to the SDA line. When the master device receives an ACK, TxE in status register 1 (R16_I2C1_STAR1) is set, and an interrupt is also generated if ITEVTEN and ITBUFEN are set. Writing data to the data register will clear the TxE bit.

If the TxE bit is set and no new data was written to the data register before the last data was sent, then the BTF bit will be set and SCL will remain low until it is cleared, and writing data to the data register after reading R16_I2C1_STAR1 will clear the BTF bit.

Figure 13-4 Master transmitter transmission sequence diagram




Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

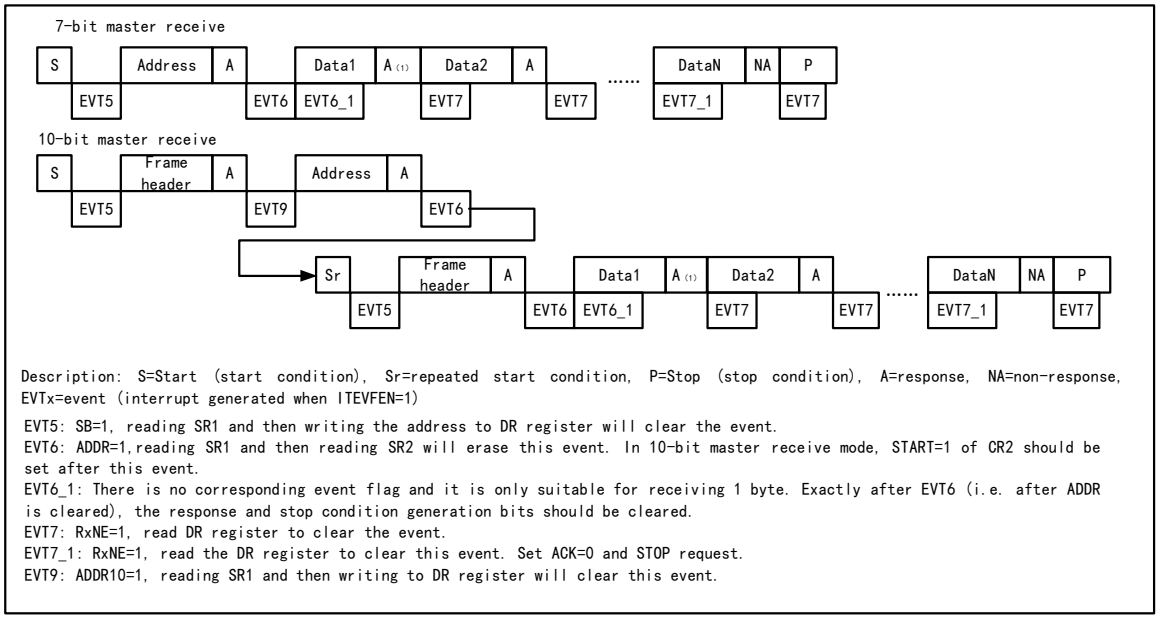
EVT5: SB=1, reading SR1 and then writing the address to the DR register will clear the event.
EVT6;ADDR=1,reading SR1 then reading SR2 will clear the event.
EVT8_1: TxE=1, shift register empty, data register empty, write DR register.
EVT8: TxE=1,shift register is not empty,data register is empty,writing DR register will clear the event.
EVT8_2: TxE=1, BTF=1, request to set the stop bit. the TxE and BTF bits are cleared by hardware when the stop condition is generated.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear the event.

*Note: 1: EVT5, EVT6, EVT9, EVT8_1 and EVT8_2 events elongate the SCL low until the end of the corresponding software sequence.*
*2: The software sequence of EVT8 must be completed before the end of the current byte transfer.*

Master receive mode:

The I2C module will receive data from the SDA line and write it into the data register via a shift register. After each byte, if the ACK bit is set, then the I2C module will send an answer low, and the RxNE bit will be set, and an interrupt will be generated if ITEVTEN and ITBUFEN are set. If RxNE is set and the original data is not read before the new data is received, the BTF bit will be set and SCL will remain low until the BTF is cleared, and reading R16_I2C1_STAR1 and then reading the data register will clear the BTF bit.

Figure 13-5 Receiver transmission sequence diagram




Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

EVT5: SB=1, reading SR1 and then writing the address to DR register will clear the event.
EVT6: ADDR=1,reading SR1 and then reading SR2 will erase this event. In 10-bit master receive mode, START=1 of CR2 should be set after this event.
EVT6_1: There is no corresponding event flag and it is only suitable for receiving 1 byte. Exactly after EVT6 (i.e. after ADDR is cleared), the response and stop condition generation bits should be cleared.
EVT7: RxNE=1, read DR register to clear the event.
EVT7_1: RxNE=1, read the DR register to clear this event. Set ACK=0 and STOP request.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear this event.

When the master device finishes sending data, it will actively send an end event, that is, set the STOP bit, and I2C will switch to the slave mode. In the receiving mode, the master device needs to NAK at the answering position of the last data bit, and after receiving the NACK, the slave device releases the control of SCL and SDA lines; The mast device can send a stop/restart condition. Note that the I2C module will automatically switch to the slave mode

after the stop condition is generated.

## 13.4 Slave Mode

In slave mode, I2C module can recognize its own address and broadcast call address. The software can control whether to enable or disable the identification of broadcast call addresses. Once the start event is detected, the I2C module compares the data of SDA with its own address (the number of bits depends on ENDUAL and ADDMODE) or broadcast address (when ENGC is set) through the shift register, and will ignore the mismatch until a new start event is generated; If it matches the header sequence, it will generate an ACK signal and wait for the address of the second byte; If the address of the second byte also matches or the address of the whole segment matches in the case of a 7-bit address, then:

Firstly, an ACK response is generated;

The ADDR bit is set, and if the ITEVTEN bit has been set, a corresponding interrupt will be generated;

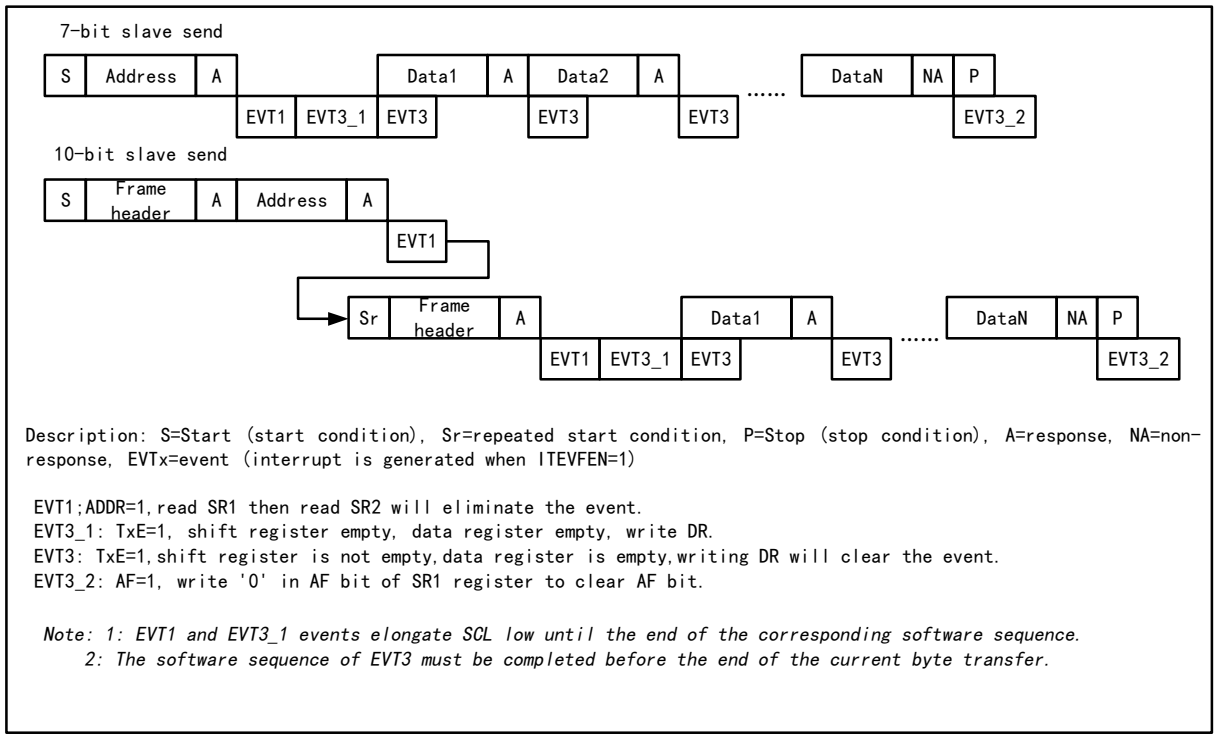
If the dual address mode is used (the ENDUAL bit is set), you need to read the DUAL bit to determine which address the host evokes.

The slave mode is receive mode by default. In case the last bit of the received header sequence is 1, or the last bit of the 7-bit address is 1 (Depending on whether the header sequence is received for the first time or a normal 7-bit address), the I2C module will go to transmitter mode and the TRA bit will indicate whether it is currently receiver or transmitter mode.

**Slave transmit mode:**

After clearing the ADDR bit, the I2C module sends bytes from the data register to the SDA line through the shift register. The slave device keeps SCL low until the ADDR bit is cleared and the data to be sent has been written into the data register. (see EVT1 and EVT3 in the figure below). After receiving a reply ACK, the TxE bit will be set, and if ITEVTEN and ITBUFEN are set, an interrupt will be generated. If TxE is set but no new data is written into the data register before the next data transmission ends, the BTF bit will be set. SCL will remain low until the BTF is cleared. After reading the status register 1(R16_I2C1_STAR1), writing data to the data register will clear the BTF bit.

Figure 13-6 Slave transmitter transmission sequence diagram




Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt is generated when ITEVFEN=1)

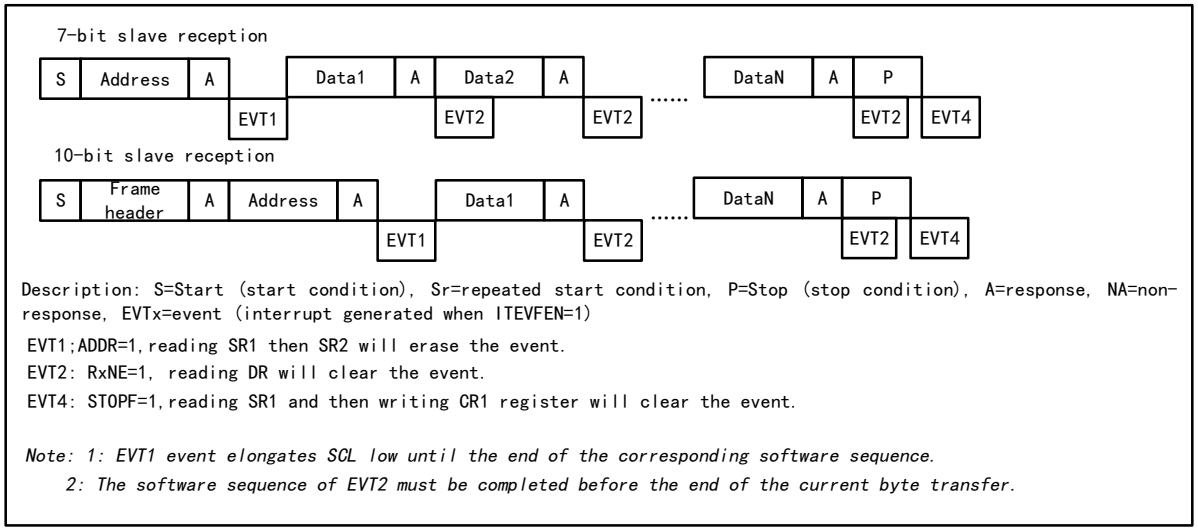
EVT1;ADDR=1,read SR1 then read SR2 will eliminate the event.
EVT3_1: TxE=1, shift register empty, data register empty, write DR.
EVT3: TxE=1,shift register is not empty,data register is empty,writing DR will clear the event.
EVT3_2: AF=1, write '0' in AF bit of SR1 register to clear AF bit.

*Note: 1: EVT1 and EVT3_1 events elongate SCL low until the end of the corresponding software sequence.*
*2: The software sequence of EVT3 must be completed before the end of the current byte transfer.*

**Slave receive mode:**

After ADDR is cleared, the I2C module stores the data on SDA into the data register via the shift register. After each byte is received, the I2C module sets an ACK bit and sets the RxNE bit, and generates an interrupt if ITEVTEN and ITBUFEN are set. If RxNE is set and the old data is not read before the new data is received, then BTF is set. SCL will remain low until the BTF bit is cleared. Reading status register 1 (R16_I2C1_STAR1) and reading the data in the data register will clear the BTF bit.

Figure 13-7 Receiver transmission sequence diagram




Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

EVT1;ADDR=1,reading SR1 then SR2 will erase the event.
EVT2: RxNE=1, reading DR will clear the event.
EVT4: STOPF=1,reading SR1 and then writing CR1 register will clear the event.

*Note: 1: EVT1 event elongates SCL low until the end of the corresponding software sequence.*
*2: The software sequence of EVT2 must be completed before the end of the current byte transfer.*

The master device will generate a stop condition after transmitting the last data byte. When the I2C module detects the stop event, it will set the STOPF bit, and if the ITEVFEN bit is set, it will also generate an interrupt. Users need to read the status register (R16_I2C1_STAR1) and then write the control register (such as the reset control word

SWRST) to clear it. (See EVT4 in the above picture).

## 13.5 Error Conditions

### 13.5.1 Bus Error (BERR)

A bus error will be generated when the I2C module detects an external start or stop event during address or data transfer. When a bus error is generated, the BERR bit is set and an interrupt is generated if ITERREN is set. In slave mode, the data is discarded and the hardware releases the bus. If it is a start signal, the hardware assumes it is a restart signal and starts waiting for an address or stop signal; if it is a stop signal, it operates ahead of normal stop conditions. In master mode, the hardware does not release the bus while not affecting the current transfer, and it is up to the user code to decide whether to abort the transfer.

### 13.5.2 Acknowledge Failure (AF)

An answer error will be generated when the I2C module detects a byte and then no answer. When an answer error is generated: AF will be set and an interrupt will be generated if ITERREN is set; when an AF error is encountered, the hardware must release the bus if the I2C module is working in slave mode and the software must generate a stop event if it is in master mode.

### 13.5.3 Arbitration Lost (ARLO)

An arbitration lost error is generated when the I2C module detects an arbitration lost. When an arbitration loss error is generated: the ARLO bit is set and an interrupt is generated if ITERREN is set; the I2C module switches to slave mode and no longer responds to transfers initiated against its slave address unless a new start event is initiated by the host; the hardware releases the bus.

### 13.5.4 Overrun/Underrun Error (OVR)

- **Overrun error**

In Slave mode, if the clock extension is disabled and the I2C module is receiving data, an overrun error will occur if a byte of data has been received but the last received data has not been read out. When an overrun error occurs, the last received byte will be discarded and the sender should retransmit the last sent byte.

- **Underrun error**

In Slave mode, if the clock is forbidden to extend and the I2C module is sending data, an underrun error will occur if new data has not been written to the data register before the next byte of the clock comes. In case of an underrun error, the data in the previous data register will be sent twice, and if an underrun error occurs, then the receiver should discard the data received repeatedly. In order not to generate an underrun error, the I2C module should write the data to the data register before the first rising edge of the next byte.

## 13.6 Clock Extension

If clock extension is disabled, then there is a possibility of overrun/underrun errors. However, if clock extension is enabled:

- In transmit mode, if TxE is set and BTF is set, SCL will always be low, always waiting for the user to read the status register and write the data to be sent to the data register.
- In receive mode, if RxNE is set and BTF is set, SCL will remain low after data is received until the user reads

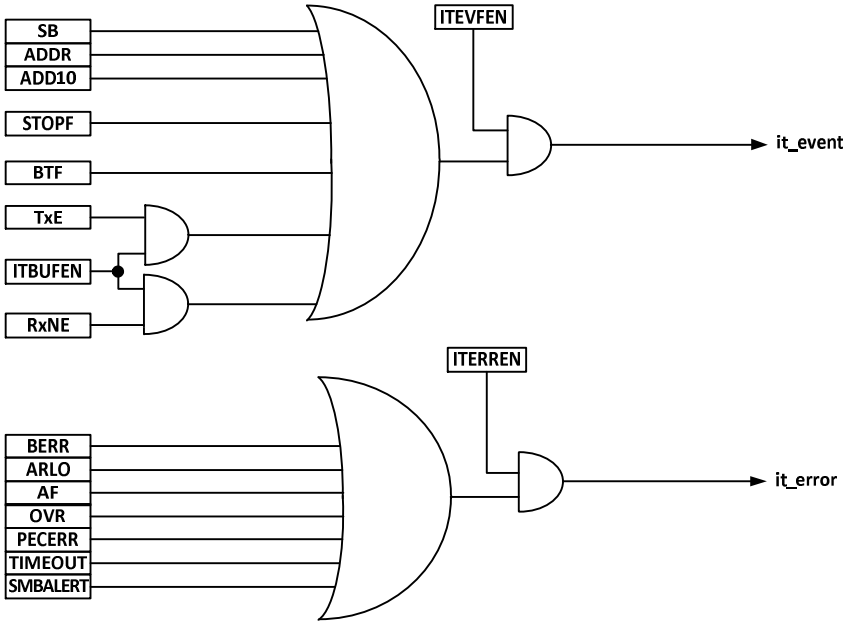
the status register and reads the data register.

It can be seen that enabling clock extension can avoid overrun/underrun errors.

## 13.7 Interrupts

Each I2C module has 2 interrupt vectors, event interrupts and error interrupts. Both interrupts support the interrupt sources in Figure 13-8.

Figure 13-8 I2C Interrupt Request




## 13.8 DMA

DMA can be used to send and receive batch data. The ITBUFEN bit of the control register cannot be set when DMA is used.

- Send by DMA

The DMA mode can be activated by setting the DMAEN bit in the CTLR2 register. As long as the TxE bit is set, data will be loaded into the I2C data register from the set memory by DMA. The following settings are required to allocate channels for I2C.

1) Set the address of I2C1_DATAR register to DMA_PADDRx register and the address of memory to DMA_MADDRx register, so that data will be sent from memory to I2C1_DATAR register after every TxE event.
2) Set the required number of transfer bytes in DMA_CNTRx register. After each TxE event, this value will be decremented.
3) Configure the channel priority by using PL[0:1] bits in DMA_CFGRx register.
4) Set the DIR bit in the DMA_CFGRx register, and according to the application requirements, it can be configured to issue an interrupt request when the whole transmission is completed halfway or completely.
5) Activate the channel by setting the EN bit in the DMA_CFGRx register.

When the number of data transmission bytes set in the DMA controller has been completed, the DMA controller sends an EOT/EOT_1 signal to the I2C interface. If the interrupt is allowed, a DMA interrupt will be generated.

- Receive by DMA

After setting DMAEN in CTLR2 register, DMA receiving mode can be performed. When receiving by DMA, DMA transfers the data in the data register to the preset memory area. The following steps are required to allocate channels for I2C.

1) Set the address of I2C1_DATAR register to DMA_PADDRx register and the address of memory to DMA_MADDRx register, so that data will be written into memory from I2C1_DATAR register after each RxNE event.
2) Set the required number of transfer bytes in DMA_CNTRx register. This value will be decremented after each RxNE event.
3) Configure the channel priority with PL[0:1] in DMA_CFGRx register.
4) Clear the DIR bit in DMA_CFGRx register, and according to the application requirements, it can be set to issue an interrupt request when the data transmission is completed halfway or completely.
5) Set the EN bit in DMA_CFGRx register to activate the channel.

When the number of data transmission bytes set in the DMA controller has been completed, the DMA controller sends an EOT/EOT_1 signal to the I2C interface. If the interrupt is allowed, a DMA interrupt will be generated.

## 13.9 Packet Error Checking

Packet Error Checksum (PEC) is an additional CRC8 checksum step to provide transmission reliability, calculated for each bit of serial data using the following polynomial.

$$C=X^8+X^2+X+1$$

The PEC calculation is activated by the ENPEC bit in the control register and is performed on all information bytes, including address and read/write bits. In transmitting, enabling PEC adds a byte of CRC8 calculation result after the last byte of data; while in receiving mode, in the last byte is considered as CRC8 check result, and if it does not match with the internal calculation result, it will reply a NAK, and in case of the main receiver, regardless of the correct check result.

## 13.10 Register Description

Table 13-1 I2C-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_I2C1_CTLR1 | 0x40005400 | I2C control register 1 | 0x0000 |
| R16_I2C1_CTLR2 | 0x40005404 | I2C control register 2 | 0x0000 |
| R16_I2C1_OADDR1 | 0x40005408 | I2C address register 1 | 0x0000 |
| R16_I2C1_OADDR2 | 0x4000540C | I2C address register 2 | 0x0000 |
| R16_I2C1_DATAR | 0x40005410 | I2C data register | 0x0000 |
| R16_I2C1_STAR1 | 0x40005414 | I2C status register 1 | 0x0000 |
| R16_I2C1_STAR2 | 0x40005418 | I2C status register 2 | 0x0000 |
| R16_I2C1_CKCFGR | 0x4000541C | I2C clock register | 0x0000 |

### 13.10.1 I2C Control Register 1 (I2C1_CTLR1)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | SWRST | RW | Software reset, setting this bit by user code will reset the I2C peripheral. Make sure the pins of the I2C bus are released and the bus is idle before the reset.<br>*Note: This bit resets the I2C module when no stop condition is detected on the bus but the busy bit is 1.* | 0 |
| [14:13] | Reserved | RO | Reserved | 0 |
| 12 | PEC | RW | Packet error checking bit, set this bit to enable packet error detection. The user code can set or clear this bit; the hardware clears this bit when the PEC is transmitted, when a start or end signal is generated, or when the PE bit is cleared to 0.<br>1: With PEC.<br>0: Without PEC.<br>*Note: The PEC is invalidated when arbitration is lost.* | 0 |
| 11 | POS | RW | ACK and PEC position setting bits, which can be set or cleared by user code and can be cleared by hardware after the PE has been cleared.<br>1: ACK bit controls the ACK or NAK of the next byte received in the shift register. The next byte received in the PEC shift register is the PEC.<br>0: ACK bit controls the ACK or NAK of the byte currently being accepted in the shift register. the PEC bit indicates that the byte in the shift register before the current bit is PEC.<br>*Note: The POS bit is used in 2-byte data reception as follows: it must be configured before reception. In order to NACK the 2nd byte, the ACK bit must be cleared immediately after clearing the ADDR bit; in order to detect the PEC of the second byte, the PEC bit must be set after the ADDR event and after configuring the POS bit.* | 0 |
| 10 | ACK | RW | Acknowledge enable, This bit is set and cleared by software and cleared by hardware when PE=0.<br>1: Acknowledge returned after a byte is received.<br>0: No acknowledge returned. | 0 |
| 9 | STOP | RW | Stop generation bit. This bit is set and cleared by software, cleared by hardware when a Stop condition is detected, set by hardware when a timeout error is detected.<br>In Master mode:<br>1: Stop generation after the current byte transfer or after the current Start condition is sent.<br>0: No Stop generation.<br>In Slave mode:<br>1: Release the SCL and SDA lines after the current byte transfer.<br>0: No Stop generation. | 0 |
| 8 | START | RW | Start generation. This bit is set and cleared by software and cleared by hardware when start is sent or PE=0. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | In Master mode:<br>1：Repeated start generation<br>0：No Start generation<br>In Slave mode:<br>1：Start generation when the bus is free<br>0：No Start generation | |
| 7 | NOSTRETCH | RW | Clock stretching disable bit. This bit is used to disable clock stretching in slave mode when ADDR or BTF flag is set, until it is reset by software.<br>1：Clock stretching disabled.<br>0：Clock stretching enabled. | 0 |
| 6 | ENGC | RW | General call enable bit. Set this bit to enable broadcast call and answer broadcast address 0x00. | 0 |
| 5 | ENPEC | RW | PEC enable bit, set this bit to enable PEC calculation. | 0 |
| [4:1] | Reserved | RO | Reserved | 0 |
| 0 | PE | RW | I2C peripheral enable bit.<br>1: Enable the I2C module.<br>0: Disable the I2C module. | 0 |

### 13.10.2 I2C Control Register 2 (I2C1_CTLR2)

Offset address: 0x04

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:13] | Reserved | RO | Reserved. | 0 |
| 12 | LAST | RW | DMA last transfer bit.<br>1: Next DMA EOT is the last transfer.<br>0: Next DMA EOT is not the last transfer.<br>*Note: This bit is used in master receiver mode to permit the generation of a NACK on the last received data.* | 0 |
| 11 | DMAEN | RW | DMA requests enable bit. Set this bit to allow DMA request when TxE or RxEN is set. | 0 |
| 10 | ITBUFEN | RW | Buffer interrupt enable bit.<br>1: When TxE or RxEN is set, event interrupt is generated.<br>0: When TxE or RxEN is set, no interrupt is generated. | 0 |
| 9 | ITEVTEN | RW | Event interrupt enable bit. Set this bit to enable event interrupt. This interrupt will be generated under the following conditions.<br>SB=1 (Master mode).<br>ADDR=1 (Master-slave mode).<br>ADDR10 = 1 (Master mode).<br>STOPF=1 (Slave mode).<br>BTF = 1, but no TxE or RxEN events.<br>TxE event to 1 if ITBUFEN = 1.<br>RxNE event to 1if ITBUFEN = 1. | 0 |
| 8 | ITERREN | RW | Error interrupt enable bit. Set to allow error interrupts.<br>The interrupt will be generated under the following conditions.<br>BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1.<br>TIMEOUT=1; SMBAlert=1. | 0 |
| [7:6] | Reserved | RO | Reserved | 0 |

| | | | | |
|---|---|---|---|---|
| [5:0] | FREQ[5:0] | RW | The I2C module clock frequency field, which must be entered at the correct clock frequency to produce the correct timing, allows a range between 8-48 MHz. It must be set between 001000b and 110000b in MHz. | 0 |

### 13.10.3 I2C Address Register 1 (I2C1_OADDR1)

Offset address: 0x08

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | ADDMODE | RW | Address mode.<br>1: 10-bit slave address (Does not respond to 7-bit addresses).<br>0: 7-bit slave address (Does not respond to 10-bit address) | 0 |
| [14:10] | Reserved | RO | Reserved | 0 |
| [9:8] | ADD[9:8] | RW | Interface address, bits 9-8 when using a 10-bit address, ignored when using a 7-bit address. | 0 |
| [7:1] | ADD[7:1] | RW | Interface address, bits 7-1. | 0 |
| 0 | ADD0 | RW | Interface address, bit 0 when using a 10-bit address, ignored when using a 7-bit address. | 0 |

### 13.10.4 I2C Address Register 2 (I2C1_OADDR2)

Offset address: 0x0C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved | 0 |
| [7:1] | ADD2[7:1] | RW | Interface address, bits 7-1 of the address in dual address mode. | 0 |
| 0 | ENDUAL | RW | Dual address mode enable bit, set this bit to allow ADD2 to be recognized as well. | 0 |

### 13.10.5 I2C Data Register (I2C1_DATAR)

Offset address: 0x10

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved | 0 |
| [7:0] | DR[7:0] | RW | Data register, this field is used to store the received data or to store the data used to send to the bus. | 0 |

### 13.10.6 I2C Status Register 1 (I2C1_STAR1)

Offset address: 0x14

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:13] | Reserved | RO | Reserved | 0 |
| 12 | PECERR | RW0 | The PEC error flag bit occurs on reception, and this bit can be reset by a user write of 0 or by hardware when PE goes low.<br>1: There is a PEC error and the PEC is received and NAK is returned.<br>0: No PEC error. | 0 |
| 11 | OVR | RW0 | Overrun and underrun flag bits.<br>1: There are overrun and underrun events occurring: when NOSTRETCH=1, when a new byte is received in receive mode, the content in the | 0 |

| | | | | |
|---|---|---|---|---|
| | | | data register has not been read out, then the newly received byte will be lost; when in send mode, no new data is written to the data register, and the same byte will be sent twice.<br>0: No overrun or underrun events. | |
| 10 | AF | RW0 | Acknowledge failure bit. Cleared by software writing 0, or by hardware when PE=0.<br>1: Acknowledge failure.<br>0: No acknowledge failure. | 0 |
| 9 | ARLO | RW0 | Arbitration loss flag bit, which can be reset by a user write of 0, or by hardware when PE goes low.<br>1: Loss of arbitration detected and the module loses control of the bus;<br>0: Arbitration normal. | 0 |
| 8 | BERR | RW0 | The bus error flag bit, which can be reset by a user write of 0, or by hardware when PE goes low.<br>1: Error in start or stop condition;<br>0: Normal. | 0 |
| 7 | TxE | RO | The data register is the empty flag bit, which can be cleared by writing data to the data register, either after generating a start or stop bit, or automatically by hardware when PE is 0.<br>1: The transmit data register is empty when data is sent;<br>0: Data register is non-empty. | 0 |
| 6 | RxNE | RO | Data register not empty bit. Cleared by software reading or writing the DR register or by hardware when PE=0.<br>1: Data register not empty.<br>0: Data register empty. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | STOPF | RO | Stop detection bit. Cleared by software reading the SR1 register followed by a write in the CR1 register, or by hardware when PE=0<br>1: After answering, the slave device detects a stop event on the bus;<br>0: No Stop condition detected. | 0 |
| 3 | ADD10 | RO | The 10-bit address header sequence sends a flag bit, which will be cleared by writing to the control register 1 after the user reads the status register 1, or by hardware when the PE is 0.<br>1: In the 10-bit address mode, the master device has sent the first address byte;<br>0: None. | 0 |
| 2 | BTF | RO | Byte sending end flag bit. After the user reads the status register 1, reading and writing the data register will clear this bit; In transmission, this bit is cleared by hardware after a start or stop event is initiated, or when PE is 0.<br>1: End of byte transmission. When NOSTRETCH=0: when sending, when a new data is sent and the data register has not been written with the new data; When receiving, when a new byte is received but the data register has not been read;<br>0: None. | 0 |
| 1 | ADDR | RW0 | Address is sent/address matching flag bit. After | 0 |

| | | | | |
|---|---|---|---|---|
| | | | the user reads status register 1, this bit will be cleared by reading status register 2, or when PE is 0, this bit will be cleared by hardware.<br>In Master mode:<br>1: End of address transmission. For 10-bit addressing, the bit is set after the ACK of the 2nd byte. For 7-bit addressing, the bit is set after the ACK of the byte.<br>0: No end of address transmission.<br>In Slave mode:<br>1: Received address matched.<br>0: Address mismatched or not received. | |
| 0 | SB | RO | The start bit transmit flag bit, an operation to write the data register after reading status register 1 will clear this bit, or the hardware will clear this bit when PE is 0.<br>1: The start bit has been transmitted;<br>0: The start bit not transmitted. | 0 |

### 13.10.7 I2C Status Register 2 (I2C1_STAR2)

Offset address: 0x18

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | PEC[7:0] | RO | Packet error check field, when PEC is enabled (ENPEC is set) this field holds the value of PEC. | 0 |
| 7 | DUALF | RO | The match detection flag bit, which is cleared by the hardware when a stop or start bit is generated, or when PE=0.<br>1: The received address matches the contents of OAR2;<br>0: The received address matches the contents in OAR1. | 0 |
| [6:5] | Reserved | RO | Reserved. | 0 |
| 4 | GENCALL | RO | The broadcast call address flag bit, which is cleared by the hardware when a stop or start bit is generated, or when PE=0.<br>1: The address of the broadcast call received when ENGC=1;<br>0: Address of broadcast call not received. | 0 |
| 3 | Reserved | RO | Reserved | 0 |
| 2 | TRA | RO | The transmit/receive flag bit, which is cleared by the hardware when a stop event is detected (STOPF=1), a repeated start condition, a bus arbitration loss (ARLO=1) or when PE=0.<br>1: Data transmitted;<br>0: Data received.<br>This bit is determined according to the RW bit of the address byte. | 0 |
| 1 | BUSY | RO | Busy flag bit, this bit will be cleared when a stop bit is detected. It is still updated when the interface is disabled (PE=0).<br>1: Bus busy: low level present in SDA or SCL;<br>0: Bus idle no communication. | 0 |
| 0 | MSL | RO | Master-slave mode indicator bit, hardware sets this bit when the interface is in master mode (SB=1); hardware clears this bit when the bus | 0 |

| | | | | |
|---|---|---|---|---|
| | | | detects a stop bit, when arbitration is lost, or when PE=0. | |

### 13.10.8 I2C Clock Register (I2C1_CKCFGR)

Offset address: 0x1C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | F/S | RW | Master mode selection bit.<br>1: Fm mode;<br>0: Sm mode. | 0 |
| 14 | DUTY | RW | Duty cycle in the fast mode:<br>1: $T_{Low\ level}/T_{High\ level}$=16/9;<br>0: $T_{Low\ level}/T_{High\ level}$=2. | 0 |
| [13:12] | Reserved | RO | Reserved | 0 |
| [11:0] | CCR[11:0] | RW | The clock division factor field, which determines the frequency waveform of the SCL clock. | 0 |

# Chapter 14 Serial Peripheral Interface (SPI)

SPI supports data interaction in a 3-wire synchronous serial mode, plus a chip selector line to support hardware switching between Master and Slave modes, and supports communication on a single data line.

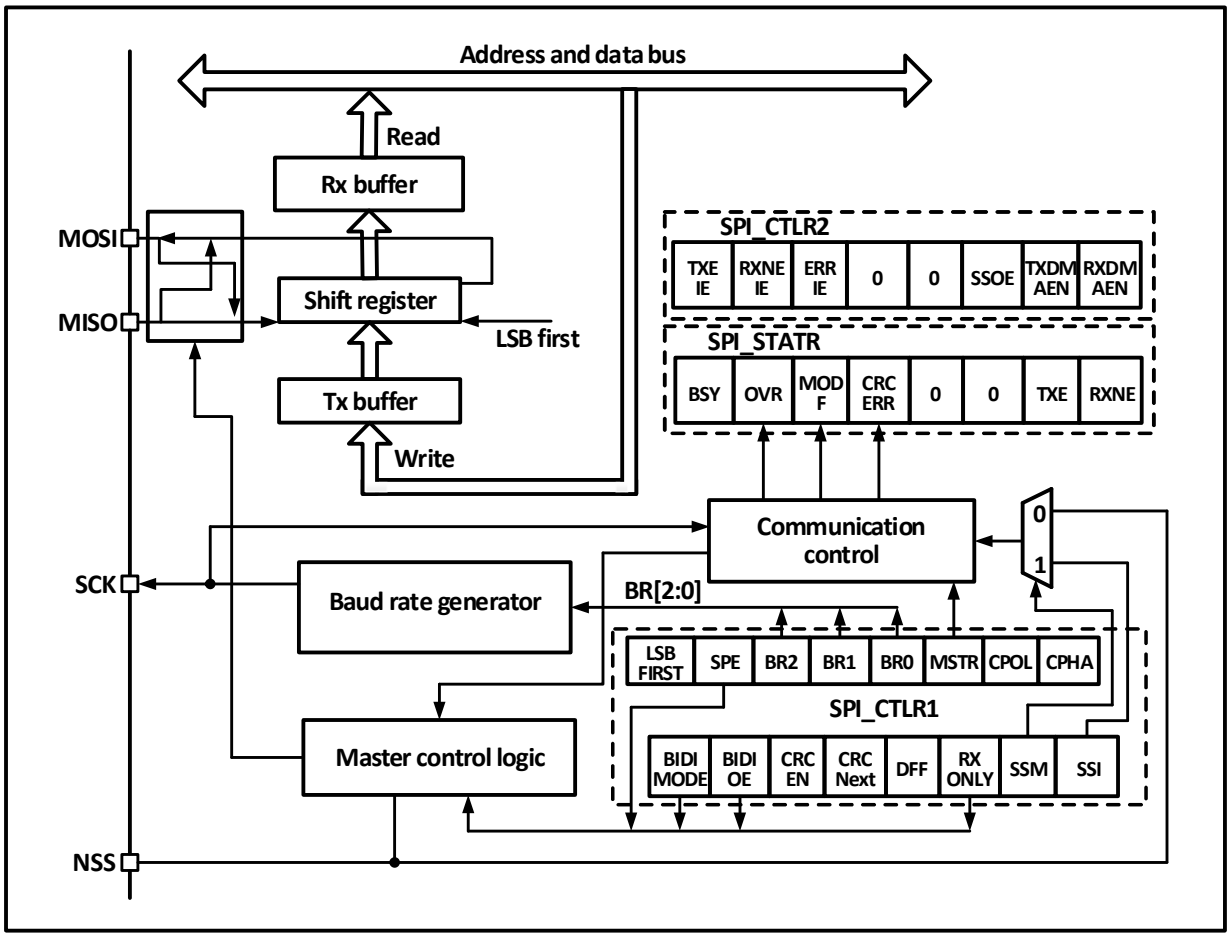
## 14.1 Main Features

- Support full-duplex synchronous serial mode
- Support single-wire half-duplex mode
- Support Master mode and Slave mode, multi-slave mode
- Support 8-bit or 16-bit data structures
- Maximum clock frequency supports up to half of $F_{HCLK}$
- Data order supports MSB or LSB first
- Support hardware or software control of NSS pins
- Transceiver supports hardware CRC check.
- Transceiver buffers support DMA transfers
- Support modification of clock phase and polarity

## 14.2 Function Description

### 14.2.1 Overview

Figure 14-1 SPI structure block diagram

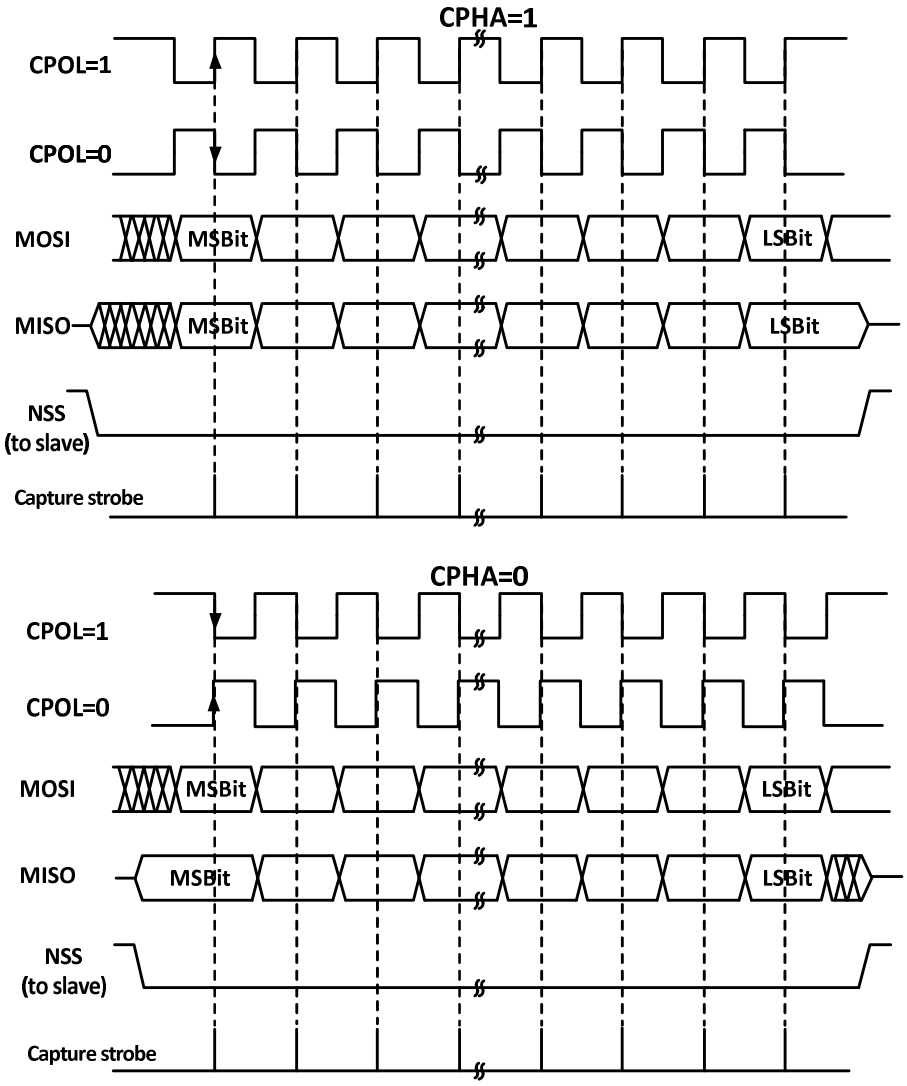



As can be seen from Figure 14-1, the 4 main SPI-related pins are MISO, M0SI, SCK and NSS. The MISO pin is the data input pin when the SPI module is operating in Master mode and the data output pin when it is operating in Slave mode. the MOSI pin is the data output pin when it is operating in Master mode and the data input pin when it is operating in Slave mode. the SCK is the clock pin, the clock signal is always output by the host and the slave receives the clock signal and synchronizes the data sending and receiving. the NSS pin is the chip select pin with the following usage:

1) NSS controlled by software: when SSM is set and the internal NSS signal is output high or low as determined by SSI, this case is generally used in SPI Master mode.
2) NSS is controlled by hardware: when NSS output is enabled, i.e., when SSOE is set, the NSS pin will be actively pulled low when the SPI host sends output outward, and a hardware error will be generated if the NSS pin is pulled low; if SSOE is not set, it can be used in Multi-master mode, and if it is pulled low it will be forced into Slave mode and the MSTR bit will be cleared automatically.

The working mode of SPI can be configured through CPHA and CPOL. When CPHA is set, it means that the module samples data at the second edge of the clock and the data is latched; when CPHA is not set, it means that the SPI module samples at the first edge of the clock and the data is latched. CPOL indicates whether the clock remains high or low when there is no data. See Figure 14-2 below for details.

Figure 14-2 SPI mode




The host and the device need to be set to the same SPI mode, and the SPE bit needs to be cleared before configuring SPI mode. The DEF bit can determine whether the single data length of SP is 8 bits or 16 bits. LSBFIRST can control whether a single data word is high-order or low-order.

### 14.2.2 Master Mode

The serial clock is generated by SCK when the SPI module is operating in master mode. The following steps are performed to configure into master mode:

1) The BR[2:0] field of the configuration control register to determine the clock;
2) Configure the CPOL and CPHA bits to determine the SPI mode;
3) Configure DEF to determine the data word length;
4) Configure LSBFIRST to determine the frame format;
5) Configure the NSS pin, for example by setting the SSOE bit and letting the hardware set the NSS. it is also possible to set the SSM bit and set the SSI bit high;
6) To set the MSTR bit and the SPE bit, you need to make sure that the NSS is already high at this time.

When data needs to be sent just write the data to be sent to the data register. SPI will send the data from the send buffer to the shift register in parallel, when the data has reached the shift register, the TXE flag will be set, if the TXEIE has been set, then an interrupt will be generated. If the TXE flag position bit needs to fill the data register with data to maintain the complete data flow.

When the receiver receives data, when the last sample clock edge of the data word comes, the data is transferred from the shift register to the receive buffer in parallel, the RXNE bit is set, and an interrupt is generated if the RXNEIE bit was previously set. At this time, the data register should be read as soon as possible to take away the data.

### 14.2.3 Slave Mode

When the SPI module is operating in slave mode, SCK is used to receive the clock from the host and its own baud rate setting is invalid. To configure into slave mode, proceed as follows.

1) Configure the DEF bit to set the data bit length.
2) Configure CPOL and CPHA bits to match the host mode;
3) Configure LSBFIRST to match the host data frame format;
4) In the hardware management mode, the NSS pin needs to be kept low. If the NSS is set to software management (SSM set), please keep the SSI not set;
5) Clear MSTR bit, set SPE bit, and start SPI mode.

When transmitting, when the first slave receives the sampling edge in SCK, the slave starts transmitting. The transmitting process is that the data in the transmitting buffer is moved to the sending shift register. When the data in the sending buffer is moved to the shift register, the TXE flag will be set. If the TXEIE bit is set before, an interrupt will be generated.

When receiving, after the last clock sampling edge, the RXNE bit is set, and the bytes received by the shift register are transferred to the receiving buffer, and the data in the receiving buffer can be obtained by reading the data register. If RXNEIE has been set before RXNE is set, an interrupt will be generated.

### 14.2.4 Simplex Mode

The SPI interface can operate in half-duplex mode, where the master device uses the MOSI pin and the slave device uses the MISO pin for communication. When using half-duplex communication, you need to set BIDIMODE and use BIDIOE to control the transmission direction.

Setting the RXONLY bit in normal full-duplex mode can put SPI module into simplex mode of receiving only. After setting RXONLY, a data pin will be released, and the pins released in master mode and slave mode are different. You can also set SPI to send only mode regardless of the received data.

### 14.2.5 CRC

The SPI module uses CRC checksum to ensure the reliability of full-duplex communication, and separate CRC calculators are used for data transmitting and receiving. the polynomial for CRC calculation is determined by the polynomial register, and different calculations are used for 8-bit data width and 16-bit data width, respectively.

Setting the CRCEN bit will enable CRC checksum and at the same time will reset the CRC calculator. After the last data byte is transmitted, setting the CRCNEXT bit will transmit the TXCRCR calculator calculation after the current byte is sent, while the CRCERR bit will be set if the last received receive shift register value does not match the

locally calculated RXCRCR calculation. Using the CRC checksum requires setting the polynomial calculator and setting the CRCEN bit when configuring the SPI operating mode, and setting the CRCNEXT bit on the last word or half-word to transmit the CRC and perform the receive CRC checksum. Note that the polynomial for the CRC calculation should be unified for both transmitting and receiving.

### 14.2.6 DMA

The SPI module supports the use of DMA to speed up data communication, either by using DMA to fill the transmit buffer or by using DMA to pick up data from the receive buffer in a timely manner. DMA will pick up or send data in a timely manner using RXNE and TXE as signals. DMA can also operate in simplex or CRC mode.

### 14.2.7 Errors

- Master mode fault (MODF)

When SPI works in the hardware management mode of NSS pin, the operation of pulling down the NSS pin externally occurs; Or in the NSS pin software management mode, the SSI bit is cleared; Or the SPE bit is cleared, causing SPI to be turned off; Or the MSTR bit is cleared, and SPI enters the slave mode. If the ERRIE bit has been set, an interrupt will be generated.

- Overrun condition

If the host sends data, but there is still unread data in the receiving buffer of the slave device, an overflow error will occur, the OVR bit will be set, and an interrupt will be generated if ERRIE is set. Send overflow error should restart the current transmission. Reading the data register and then reading the status register will clear this bit.

- CRC error

When the received CRC word and the value of RXCRCR do not match, a CRC error will be generated and the CRCERR bit will be set.

### 14.2.8 Interrupts

The interrupt of SPI module supports five interrupt sources, in which two events, the sending buffer is empty and the receiving buffer is not empty, will set TXE and RXNE respectively, and an interrupt will be generated when the TXEIE and RXNEIE bits are set respectively. In addition, the three errors mentioned above will also generate interrupts, namely MODF, OVR and CRCERR. After the ERRIE bit is enabled, these three errors will also generate errors.

## 14.3 Register Description

Table 14-2 SPI-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_SPI1_CTLR1 | 0x40013000 | SPI control register 1 | 0x0000 |
| R16_SPI1_CTLR2 | 0x40013004 | SPI control register 2 | 0x0000 |
| R16_SPI1_STATR | 0x40013008 | SPI status register | 0x0002 |
| R16_SPI1_DATAR | 0x4001300C | SPI data register | 0x0000 |
| R16_SPI1_CRCR | 0x40013010 | SPI polynomial register | 0x0007 |
| R16_SPI1_RCRCR | 0x40013014 | SPI receive CRC register | 0x0000 |
| R16_SPI1_TCRCR | 0x40013018 | SPI transmit CRC register | 0x0000 |
| R16_SPI1_HSCR | 0x40013024 | SPI high-speed control register | 0x0000 |

### 14.3.1 SPI Control Register 1 (SPI1_CTLR1)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | BIDIMODE | RW | Unidirectional data mode enable bit:<br>1: Select 1-wire bidirectional mode;<br>0: Select 2-wire bidirectional mode. | 0 |
| 14 | BIDIOE | RW | Single-wire output enable bit, used in conjunction with BIDIMODE.<br>1: Enable output, transmit only;<br>0: Disable output, receive only. | 0 |
| 13 | CRCEN | RW | Hardware CRC checksum enable bit, this bit can only be written when SPE is 0. This bit can only be used in full-duplex mode.<br>1: Enable CRC calculation;<br>0: Disable CRC calculation. | 0 |
| 12 | CRCNEXT | RW | Transmit the value of the CRC register after the next data transfer. This should be set immediately after the last data is written to the data register.<br>1: Transmit the result of the CRC checksum;<br>0: Continue transmitting data from the data register. | 0 |
| 11 | DFF | RW | Data frame length bit, this bit can only be written when SPE is 0.<br>1: Use the 16-bit data length for sending and receiving;<br>0: Use 8-bit data length for sending and receiving. | 0 |
| 10 | RXONLY | RW | Receive only bit in 2-wire mode, this bit is used in conjunction with BIDIMODE. Set this bit to allow the device to receive only and not transmit.<br>1: Receive only, simplex mode;<br>0: Full-duplex mode. | 0 |
| 9 | SSM | RW | Chip select pin management bit, this bit determines whether the level of the NSS pin is controlled by hardware or software.<br>1: Software control of the NSS pin;<br>0: Hardware control of the NSS pin. | 0 |
| 8 | SSI | RW | Chip select pin control bit, with SSM set, this bit determines the level of the NSS pin.<br>1: NSS is high;<br>0: NSS is low. | 0 |
| 7 | LSBFIRST | RW | Frame format control bit. It is not possible to modify this bit during communication.<br>1: LSB is transmitted first;<br>0: MSB is transmitted first. | 0 |
| 6 | SPE | RW | SPI enable bit.<br>1: SPI enabled;<br>0: SPI disabled. | 0 |
| [5:3] | BR[2:0] | RW | Modified during communication.<br>000: FHCLK/2; 001: FHCLK/4;<br>010: FHCLK/8; 011: FHCLK/16;<br>100: FHCLK/32; 101: FHCLK/64;<br>110: FHCLK/128; 111: FHCLK/256.<br>*Note: This bit applies only if the HSRXEN bit is 0; when the HSRXEN bit is 1, the SCK frequency is FHCLK/(BR+2).* | 0 |
| 2 | MSTR | RW | Master-slave setting bit, this bit cannot be modified during communication. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Configured as a master device;<br>0: Configured as a slave device. | |
| 1 | CPOL | RW | Clock polarity selection bit, this bit must not be modified during communication.<br>1: SCK is held high in the idle state;<br>0: SCK is held low in the idle state.<br>*Note: When using SPI to send data from mode, this bit should be set to 1, and mode 2 or mode 3 should be selected.* | 0 |
| 0 | CPHA | RW | Clock phase set bit, this bit must not be modified during communication.<br>1: Data sampling starts from the second clock edge;<br>0: Data sampling starts from the first clock edge. | 0 |

### 14.3.2 SPI Control Register 2 (SPI1_CTLR2)

Offset address: 0x04

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved | 0 |
| 7 | TXEIE | RW | Transmit buffer air break enable bit. Set this bit to allow an interrupt to be generated when TXE is set. | 0 |
| 6 | RXNEIE | RW | Receive buffer non-air break enable bit. Set this bit to allow an interrupt to be generated when RXNE is set. | 0 |
| 5 | ERRIE | RW | Error interrupt enable bit. Set this bit to allow an interrupt to be generated if an error (CRCERR, OVR, MODF) is generated. | 0 |
| [4:3] | Reserved | RO | Reserved | 0 |
| 2 | SSOE | RW | SS output enable. Disabling SS output can work in multi-master mode.<br>1: Enable the SS output;<br>0: Disable SS output in master mode. | 0 |
| 1 | TXDMAEN | RW | Transmit buffer DMA enable bit.<br>1: Enable transmit buffer DMA;<br>0: Disable transmit buffer DMA. | 0 |
| 0 | RXDMAEN | RW | Receive buffer DMA enable bit.<br>1: Enable receive buffer DMA;<br>0: Disable receive buffer DMA. | 0 |

### 14.3.3 SPI Status Register (SPI1_STATR)

Offset address: 0x08

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved | 0 |
| 7 | BSY | RO | Busy flag bit, this bit is set or reset by hardware.<br>1: SPI is communicating, or the transmit buffer is not empty;<br>0: SPI is not communicating. | 0 |
| 6 | OVR | RW0 | Overflow flag bit, this bit is set by hardware and reset by software.<br>1: An overflow error has occurred;<br>0: No overflow error has occurred. | 0 |
| 5 | MODF | RO | Mode error flag bit, this bit is set by hardware and reset by software.<br>1: A mode error has occurred;<br>0: No mode error has occurred. | 0 |
| 4 | CRCERR | RW0 | CRC error flag bit, this bit is set by hardware and | 0 |

| | | | | |
|---|---|---|---|---|
| | | | reset by software.<br>1: The received CRC value does not match the value of the RCRCR;<br>0: The received CRC value is the same as the value of the RRCR. | |
| [3:2] | Reserved | RO | Reserved | 0 |
| 1 | TXE | RO | Transmit buffer empty flag bit:<br>1: The transmit buffer is empty;<br>0: The transmit buffer is not empty. | 1 |
| 0 | RXNE | RO | Receive buffer not empty flag bit:<br>1: Receive buffer is not empty;<br>0: Receive buffer is empty.<br>*Note: Read DATAR, and it will be cleared automatically.* | 0 |

### 14.3.4 SPI Data Register (SPI1_DATAR)

Offset address: 0x0C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | DR[15:0] | RW | Data register. The data registers are used to store received data or pre-store data to be sent out, so the reading and writing of data registers actually corresponds to the operation of different areas, where reads correspond to the receive buffer and writes to the send buffer. Data can be received and sent in 8 or 16 bits, and it is necessary to determine how many bits of data to use before transmission. When using 8 bits for data transmission, only the lower 8 bits of the data register are used and the higher 8 bits are forced to 0 for reception; using a 16-bit data structure causes all 16 bits of the data register to be used. | 0 |

### 14.3.5 SPI Polynomial Register (SPI1_CRCR)

Offset address: 0x10

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | CRCPOLY[15:0] | RW | CRC polynomial. This field defines the polynomial used in the CRC calculation. | 0x0007 |

### 14.3.6 SPI Receive CRC Register (SPI1_RCRCR)

Offset address: 0x14

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | RXCRC[15:0] | RO | Receive CRC value. Stores the result of the calculated CRC checksum of the received byte. Setting CRCEN resets this register. The calculation uses the polynomial used by CRCPOLY; only the lower 8 bits are involved in the calculation in 8-bit mode and all 16 bits are involved in the calculation in 16-bit mode. This register needs to be read when BSY is 0. | 0 |

### 14.3.7 SPI Transmit CRC Register (SPI1_TCRCR)

Offset address: 0x18

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | TXCRC[15:0] | RO | Transmit CRC value. Stores the result of the calculated CRC checksum of the bytes that have been sent out. Setting CRCEN resets this register. The calculation is done using the polynomial used in CRCPOLY. 8-bit mode only the lower 8 bits are involved in the calculation, 16-bit mode all 16 bits are involved. This register needs to be read when BSY is 0. | 0 |

### 14.3.8 SPI High-speed Control Register (SPI1_HSCR)

Offset address: 0x24

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:1] | Reserved | RO | Reserved | 0 |
| 0 | HSRXEN | WO | Read enable in SPI high speed mode:<br>1: Enable high-speed read mode;<br>0: Disable high-speed read mode. | 0 |

# Chapter 15 USB PD Controller (USBPD)

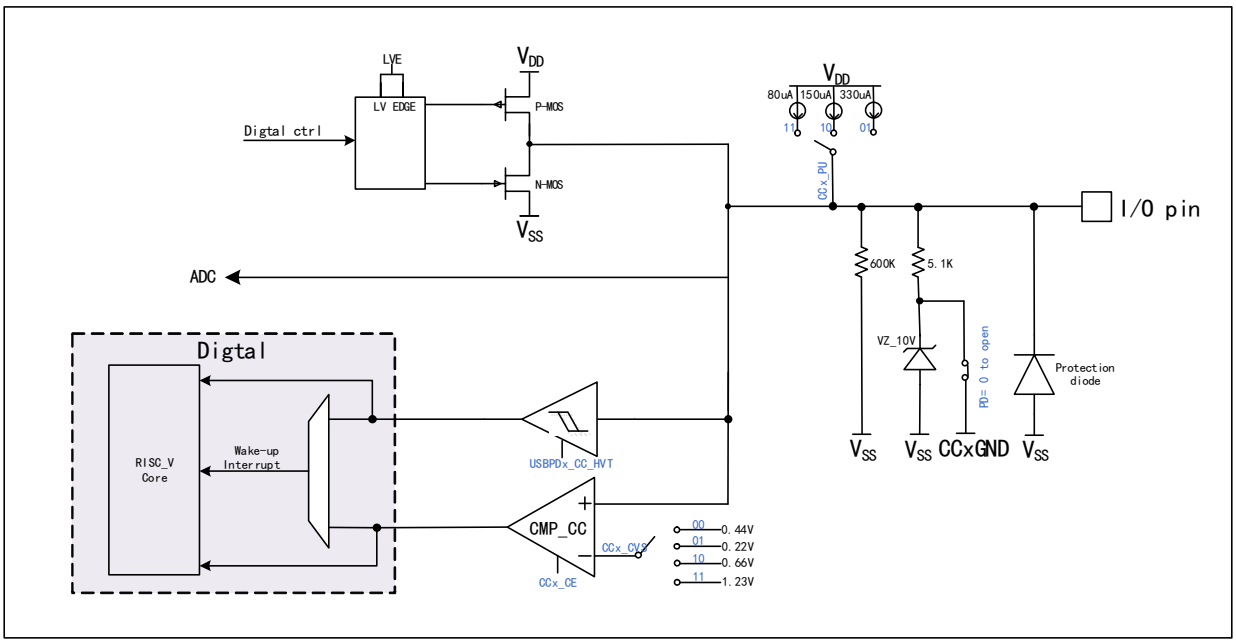
## 15.1 Introduction to USB PD Controller

The chip has built-in 2 groups of USB Power Delivery controllers and PD transceiver PHYs, providing 4 high voltage resistant CC pins.

It supports USB Type-C master-slave detection, automatic BMC codec and CRC, hardware edge control, USB PD2.0 and PD3.0 power delivery control, fast charging, UFP/PD powered end Sink and DFP/PD powered end Source application, DRP application and dynamic switching. Built-in USB Type-C interface, support master-slave detection, DRP, Sink/Consumer and Source/Provider.

- Built-in USBPD transceiver PHY, integrated hardware edge slope control
- Built-in USB Power Delivery controller, automatic BMC codec, 4b5b codec and CRC.
- Support PD packages such as SOP, SOP' and SOP", and support hardware reset of USB PD reset signal frame.
- Support the maximum packet length of 510 bytes, and support DMA.
- Support USB PD 2.0 and 3.0 power transmission protocols, and USB port supports charging protocols such as BC.
- The comparator is supported, and the negative input reference voltage can be selected in multiple steps, programmable pull-up current.
- Support programmable charging current module ISINK cooperation can support PPS high-precision voltage regulation.

Figure 15-1 CC port structure diagram




*Note: CCxGND is the internal GND of CCx. If CCx has suffix R, that is CCxR, its CCxGND has been shorted to $V_{SS}$ inside the chip.*

## 15.2 Register Description

Table 15-1 USBPD-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_USBPD_CONFIG | 0x40024000 | PD configuration register | 0x0000000X |
| R16_CONFIG | 0x40024000 | PD interrupt enable register | 0x000X |
| R16_BMC_CLK_CNT | 0x40024002 | BMC sampling clock counter | 0x0000 |
| R32_USBPD_CONTROL | 0x40024004 | PD control register | 0x00000000 |
| R16_CONTROL | 0x40024004 | PD transceiver control register | 0x0000 |
| R8_CONTROL | 0x40024004 | PD transceiver enable register | 0x00 |
| R8_TX_SEL | 0x40024005 | PD transmit SOP select register | 0x00 |
| R16_BMC_TX_SZ | 0x40024006 | PD transmit length register | 0x0000 |
| R32_USBPD_STATUS | 0x40024008 | PD status register | 0x000000XX |
| R16_STATUS | 0x40024008 | PD interrupt and data register | 0x00XX |
| R8_DATA_BUF | 0x40024008 | DMA cache data register | 0xXX |
| R8_STATUS | 0x40024009 | PD interrupt flag register | 0x00 |
| R16_BMC_BYTE_CNT | 0x4002400A | Byte counter | 0x0000 |
| R32_USBPD_PORT | 0x4002400C | Port control register | 0x00030003 |
| R8_PORT_CC1 | 0x4002400C | CC1 port control register | 0x03 |
| R8_PORT_CC2 | 0x4002400E | CC2 port control register | 0x03 |
| R32_USBPD_DMA | 0x40024010 | DMA cache address register | 0x0000XXXX |
| R16_DMA | 0x40024010 | PD buffer start address register | 0x0XXX |

Table 15-2 USBPD1-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_USBPD_CONFIG | 0x40024400 | PD configuration register | 0x0000000X |
| R16_CONFIG | 0x40024400 | PD interrupt enable register | 0x000X |
| R16_BMC_CLK_CNT | 0x40024402 | BMC sampling clock counter | 0x0000 |
| R32_USBPD_CONTROL | 0x40024404 | PD control register | 0x00000000 |
| R16_CONTROL | 0x40024404 | PD transceiver control register | 0x0000 |
| R8_CONTROL | 0x40024404 | PD transceiver enable register | 0x00 |
| R8_TX_SEL | 0x40024405 | PD transmit SOP select register | 0x00 |
| R16_BMC_TX_SZ | 0x40024406 | PD transmit length register | 0x0000 |
| R32_USBPD_STATUS | 0x40024408 | PD status register | 0x000000XX |
| R16_STATUS | 0x40024408 | PD interrupt and data register | 0x00XX |
| R8_DATA_BUF | 0x40024408 | DMA cache data register | 0xXX |
| R8_STATUS | 0x40024409 | PD interrupt flag register | 0x00 |
| R16_BMC_BYTE_CNT | 0x4002440A | Byte counter | 0x0000 |
| R32_USBPD_PORT | 0x4002440C | Port control register | 0x00030003 |
| R8_PORT_CC1 | 0x4002440C | CC1 port control register | 0x03 |
| R8_PORT_CC2 | 0x4002440E | CC2 port control register | 0x03 |
| R32_USBPD_DMA | 0x40024410 | DMA cache address register | 0x0000XXXX |
| R16_DMA | 0x40024410 | PD buffer start address register | 0x0XXX |

### 15.2.1 PD Configuration Register (R32_USBPD_CONFIG)

Offset address: 0x00

| Bit | Name |
|---|---|
| [31:16] | R16_BMC_CLK_CNT |
| [15:0] | R16_CONFIG |

### 15.2.2 PD Interrupt Enable Register (R16_CONFIG)

Offset address: 0x00

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | IE_TX_END | RW | Transmit end interrupt enable. | 0 |
| 14 | IE_RX_RESET | RW | Receive reset interrupt enable. | 0 |
| 13 | IE_RX_ACT | RW | Receive complete interrupt enable. | 0 |
| 12 | IE_RX_BYTE | RW | Receive byte interrupt enable. | 0 |
| 11 | IE_RX_BIT | RW | Receive bit interrupt enable. | 0 |
| 10 | IE_PD_IO | RW | PD IO interrupt enable. | 0 |
| 9 | Reserved | RO | Reserved. | 0 |
| 8 | RX_MULTI_0 | RO | Continuous multi-bit 0 indication bits are received:<br>1: Continuous multi-bit 0 received;<br>0: Consecutive multi-bit 0s not received. | 0 |
| [7:6] | Reserved | RO | Reserved. | 0 |
| 5 | WAKE_POLAR | RW | PD port wake-up level:<br>1: Active high;<br>0: Active low. | 0 |
| 4 | PD_RST_EN | RW | PD mode reset command enable:<br>1: Reset;<br>0: Invalid. | 0 |
| 3 | PD_DMA_EN | RW | Enable DMA for USBPD, this bit must be set to 1 in normal transmission mode:<br>1: Enable DMA function and DMA interrupt;<br>0: Disable DMA. | 0 |
| 2 | CC_SEL | RW | Select the current PD communication port:<br>1: Use CC2 port for communication;<br>0: Use CC1 port for communication. | 0 |
| 1 | PD_ALL_CLR | RW | PD mode clears all interrupt flag bits: 0: invalid;<br>1: use CC2 port<br>1: Clear the interrupt flag bits;<br>0: Invalid. | 1 |
| 0 | CC_FILTER_EN | RW | Control the input filter enable on the PD pin:<br>1: Turn on filtering;<br>0: Turn off filtering; | X |

### 15.2.3 BMC Sample Clock Counter (R16_BMC_CLK_CNT)

Offset address: 0x02

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:9] | Reserved | RO | Reserved | 0 |
| [8:0] | BMC_CLK_CNT[8:0] | RW | The BMC transmits or receives a sample clock counter. | 0 |

### 15.2.4 PD Control Register (R32_USBPD_CONTROL)

Offset address: 0x04

| Bit | Name |
|---|---|
| [31:16] | R16_BMC_TX_SZ |
| [15:0] | R16_CONTROL |

### 15.2.5 PD Transceiver Control Register (R16_CONTROL)

Offset address: 0x04

| Bit | Name |
|---|---|
| [15:8] | R8_TX_SEL |

| | |
|---|---|
| [7:0] | R8_CONTROL |

### 15.2.6 PD Transceiver Enable Register (R8_CONTROL)

Offset address: 0x04

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | BMC_BYTE_HI | RO | Indicates the current half-byte status of the PD data being sent and received:<br>0: The lower 4 bits are being processed;<br>1: Indicates that the high 4 bits are being | 0 |
| 6 | TX_BIT_BACK | RO | Indicates the current bit status of the BMC when sending the code:<br>1: Indicate that a BMC byte is being transmitted;<br>0: Idle. | 0 |
| 5 | DATA_FLAG | RO | Cache data valid flag bit. | 0 |
| [4:2] | RX_STATE[2:0] | RO | PD receive status identification<br>000: Receive initial status; 001: Start receiving SOP;<br>010: Receive reset; 011: Receive SOP;<br>100: Receive end; 101: Receive unused;<br>110: Receive EOP; 111: Receive byte. | 0 |
| 1 | BMC_START | RW | BMC transmit start signal. | 0 |
| 0 | PD_TX_EN | RW | USBPD receive and transmit mode and transmit enable:<br>0: PD receive enable;<br>1: PD transmit enable. | 0 |

### 15.2.7 PD Transmit SOP Selection Register (R8_TX_SEL)

Offset address: 0x05

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | TX_SEL4[1:0] | RW | K-CODE4 type selection in PD transmit mode:<br>00: SYNC2;<br>01: SYNC3;<br>1x: RST2. | 0 |
| [5:4] | TX_SEL3[1:0] | RW | K-CODE3 type selection in PD transmit mode:<br>00: SYNC1;<br>01: SYNC3;<br>1x: RST1. | 0 |
| [3:2] | TX_SEL2[1:0] | RW | K-CODE2 type selection in PD transmit mode:<br>00: SYNC1;<br>01: SYNC3;<br>1x: RST1. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | TX_SEL1 | RW | K-CODE1 type selection in PD transmit mode:<br>0: SYNC1;<br>1: RST1. | 0 |

### 15.2.8 PD Transmit Length Register (R16_BMC_TX_SZ)

Offset address: 0x06

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:9] | Reserved | RO | Reserved | 0 |
| [8:0] | BMC_TX_SZ[8:0] | RW | The total length sent in PD mode. | 0 |

### 15.2.9 Status Register (R32_USBPD_STATUS)

Offset address: 0x08

| Bit | Name |
|---|---|
| [31:16] | R16_BMC_BYTE_CNT |
| [15:0] | R16_STATUS |

### 15.2.10 PD Interrupt and Data Register (R16_STATUS)

Offset address: 0x08

| Bit | Name |
|---|---|
| [15:8] | R8_STATUS |
| [7:0] | R8_DATA_BUF |

### 15.2.11 DMA Buffer Data Register (R8_DATA_BUF)

Offset address: 0x08

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | DATA_BUF | RO | DMA buffer data. | X |

### 15.2.12 PD Interrupt Flag Register (R8_STATUS)

Offset address: 0x09

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | IF_TX_END | RW1 | Transmit complete interrupt flag, write 1 to clear 0, write 0 is invalid. | 0 |
| 6 | IF_RX_RESET | RW1 | Receive reset interrupt flag, write 1 to clear 0, write 0 is invalid. | 0 |
| 5 | IF_RX_ACT | RW1 | Receive complete interrupt flag, write 1 to clear 0, write 0 is invalid. | 0 |
| 4 | IF_RX_BYTE | RW1 | Receive byte or SOP interrupt flag, write 1 to clear 0, write 0 is invalid. | 0 |
| 3 | IF_RX_BIT | RW1 | Receive bit or 5bit interrupt flag, write 1 to clear 0, write 0 is invalid. | 0 |
| 2 | BUF_ERR | RW1 | BUFFER or DMA error interrupt flag, write 1 to clear 0, write 0 is invalid. | 0 |
| [1:0] | BMC_AUX[1:0] | RO | Indicates the current PD status:<br>During PD reception, the status is as follows:<br>00: reception idle or no valid packet received;<br>01: SOP received i.e. SOP0;<br>10: SOP' received i.e. SOP1 or Hard Reset;<br>11: SOP' received i.e. SOP2 or Cable Reset.<br>When PD is sending, the status is as follows:<br>00: CRC32[7:0] is being transmitted;<br>01: CRC32[15:8] is being transmitted;<br>10: CRC32[23:16] is being transmitted;<br>11: CRC32[31:24] is being transmitted. | 0 |

### 15.2.13 Byte Counter (R16_BMC_BYTE_CNT)

Offset address: 0x0A

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:9] | Reserved | RO | Reserved. | 0 |
| [8:0] | BMC_BYTE_CNT[8:0] | RO | Byte counter. | 0 |

### 15.2.14 Port Control Register (R32_USBPD_PORT)

Offset address: 0x0C

| Bit | Name |
|---|---|
| [31:24] | R8_PORT_CC3 |
| [23:16] | R8_PORT_CC2 |
| [15:8] | Reserved |
| [7:0] | R8_PORT_CC1 |

### 15.2.15 CC1 Port Control Register (R16_PORT_CC1)

Offset address: 0x0C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | CC1_CE | RW | Enable the CC1 port voltage CMP:<br>1: Enable the CMP, select the reference voltage by CC1_CVS;<br>0: Disable the CMP, default CC1_CMPO output 1. | 0 |
| [6:5] | CC1_CVS[1:0] | RW | Select the reference voltage of CC1 port voltage CMP:<br>00: 0.55V;<br>01: 0.22V;<br>10: 0.66V;<br>11: 1.23V. | 0 |
| 4 | CC1_LVE | RW | CC1 port output low voltage enable:<br>1: PD low voltage drive output;<br>0: Normal $V_{DD}$ voltage drive output. | 0 |
| [3:2] | CC1_PU[1:0] | RW | CC1 port pull-up current selection (need to enable CC1_REF bit)<br>00: Disable pull-up current;<br>01: Rated 330μA;<br>10: Rated 180μA;<br>11: Rated 80μA. | 0 |
| 1 | CC1_PD | RW | CC1 port pull-down resistor Rd enable:<br>1: Enable the Rd pull-down resistor, about 5.1KΩ;<br>0: Disable the pull-down resistor.<br>*Note: There is still a weak pull-down of about 600kΩ after turning off.* | 1 |
| 0 | CC1_CMPO | RO | The result output bits of the CC1 voltage CMP:<br>1: The current port voltage is greater than the reference voltage;<br>0: The current port voltage is less than the reference voltage. | 1 |

*Note: When using the USBPD function, the ISINKEN bit needs to be turned on.*

### 15.2.16 CC2 Port Control Register (R16_PORT_CC2)

Offset address: 0x0E

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | CC2_CE | RW | Enable the CC2 port voltage CMP:<br>1: Enable the CMP and select the reference voltage by CC2_CVS;<br>0: Disable the CMP, default CC2_CMPO output 1. | 0 |
| [6:5] | CC2_CVS[1:0] | RW | Select the reference voltage of the CC2 port voltage CMP:<br>00: 0.55V; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 01: 0.22V;<br>10: 0.66V;<br>11: 1.23V. | |
| 4 | CC2_LVE | RW | CC2 port output low voltage enable:<br>1: PD low voltage drive output;<br>0: Normal $V_{DD}$ voltage drive output. | 0 |
| [3:2] | CC2_PU | RW | CC2 port pull-up current selection (need to enable CC2_REF bit)<br>00: Pull-up current disable;<br>01: 330μA;<br>10: 180μA;<br>11: 80μA. | 0 |
| 1 | CC2_PD | RW | CC2 port pull-down resistor Rd is enabled:<br>1: Enable the Rd pull-down resistor, about 5.1KΩ;<br>0: Disable the pull-down resistor<br>*Note: There is still a weak pull-down of about 600kΩ after turning off.* | 1 |
| 0 | CC2_CMPO | RO | The result output bits of the CC2 voltage CMP:<br>1: The current port voltage is greater than the reference voltage;<br>0: The current port voltage is less than the reference voltage. | 1 |

*Note: When using the USBPD function, the ISINKEN bit needs to be turned on.*

### 15.2.17 DMA Cache Address Register (R32_USBPD_DMA)

Offset address: 0x10

| Bit | Name |
|---|---|
| [31:16] | Reserved |
| [15:0] | R16_DMA |

### 15.2.18 PD Buffer Start Address Register (R16_DMA)

Offset address: 0x10

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | Reserved | RO | Reserved | 0 |
| [11:0] | USBPD_DMA_ADDR | RW | USBPD_DMA cache address.<br>The lower 12 bits are valid and the address must be 4-byte aligned. | X |

# Chapter 16 USB Full-speed Host/Device Controller (USBFS)

## 16.1 Introduction to USB Controller

Embedded with USB2.0 controller and USB-PHY, it has dual roles of host controller and USB device controller. When used as a host controller, it can support low-speed and full-speed USB devices/HUB. When used as a device controller, it can be flexibly set to low-speed and full-speed mode to adapt to various applications.

The chip is embedded with a USB controller and transceiver with the following features:

- Dual role controller supporting USB Host function and USB Device function.
- Both host and device modes support USB2.0 full-speed 12Mbps or low-speed 1.5Mbps.
- Support software HNP and SRP protocols, support PDUSB.
- Support USB control transfers, bulk transfers, interrupt transfers, and synchronous/real-time transfers.
- Support packets up to 64 bytes, built-in FIFO, interrupt and DMA support.
- Support BC1.2 and HV DCP/CDP and other high-voltage charging protocols.
- Built-in 6-bit DAC and output buffer support input comparison
- Support programmable voltage output, programmable pull-up and pull-down resistors

*Note: Refer to Chapter 20 for the BC interface.*

## 16.2 Register Description

USB related registers are divided into 3 sections, some of which are alternate in host and device mode.

- USB Global Register
- USB Device Control Register
- USB Host Control Register

### 16.2.1 Global Register Description

Table 16-1 USBFS-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_BASE_CTRL | 0x40023400 | USB control register | 0x06 |
| R8_INT_EN | 0x40023402 | USB interrupt enable register | 0x00 |
| R8_DEV_ADDR | 0x40023403 | USB device address register | 0x00 |
| R8_MIS_ST | 0x40023405 | USB miscellaneous status register | 0xXX |
| R8_INT_FG | 0x40023406 | USB interrupt flag register | 0x20 |
| R8_INT_ST | 0x40023407 | USB interrupt status register | 0xXX |
| R16_RX_LEN | 0x40023408 | USB receive length register | 0xXX |

#### 16.2.1.1 USB Control Register (R8_BASE_CTRL)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UC_HOST_MODE | RW | USB operating mode selection bits:<br>1: Host mode (HOST);<br>0: Device mode (DEVICE). | 0 |
| 6 | RB_UC_LOW_SPEED | RW | USB bus signal transmission rate selection bit:<br>1: 1.5Mbps; 0: 12Mbps. | 0 |
| 5 | RB_UC_DEV_PU_EN | RW | The USB device enable and internal pull-up resistor control bits in USB device mode, a | 0 |

| | | | | |
|---|---|---|---|---|
| | | | value of 1 enables USB device transfer and enables the internal pull-up resistor. | |
| [5:4] | MASK_UC_SYS_CTRL[1:0] | RW | See the table below to configure USB system. | 0 |
| 3 | RB_UC_INT_BUSY | RW | USB transfer completion interrupt flag not cleared to zero Auto pause enable bit:<br>1: Auto pause before interrupt flag UIF_TRANSFER is cleared, auto answer busy NAK in device mode, auto pause subsequent transfers in host mode;<br>0: No pause. | 0 |
| 2 | RB_UC_RST_SIE | RW | USB protocol processor software reset control bits:<br>1: Forced reset of the USB protocol processor (SIE), requiring a software reset;<br>0: No reset. | 1 |
| 1 | RB_UC_CLR_ALL | RW | The FIFO and interrupt flag of USB are cleared:<br>1. Forced clearing and clearing;<br>2. 0: unclear. | 1 |
| 0 | RB_UC_DMA_EN | RW | Enables DMA for USB, this bit must be set to 1 in normal transfer mode:<br>1: Enable the DMA function and the DMA interrupt;<br>0: Disable DMA. | 0 |

When RB_UC_RST_SIE=0, the USB system control combination consists of RB_UC_HOST_MODE and MASK_UC_SYS_CTRL:

Table 16-2 USB system control combination

| RB_UC_HOST_MODE | MASK_UC_SYS_CTRL | USB system control description |
|---|---|---|
| 0 | 00 | Disable USB device function, turn off internal pull-up resistor. |
| 0 | 01 | Enable USB device function, turn off internal pull-up resistor, and add external pull-up. |
| 0 | 1x | Enable USB device function and internal 1.5K pull-up resistor. This pull-up resistor takes precedence over the pull-down resistor and can also be used in GPIO mode. |
| 1 | 00 | USB host mode, working normally. |
| 1 | 01 | USB host mode, forcing DP/DM to output SE0 status. |
| 1 | 10 | USB host mode, forcing DP/DM to output J state. |
| 1 | 11 | USB host mode, forcing DP/DM to output K status/wake-up. |

### 16.2.1.2 USB Interrupt Enable Register (R8_USB_INT_EN)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved. | 0 |
| 6 | RB_UIE_DEV_NAK | RW | USB device mode, receive NAK interrupt:<br>1: Enable interrupt;<br>0: Disable interrupt. | 0 |
| 5 | RB_UID_1_WIRE | RW | USB 1-wire mode enable:<br>1: Enable; 0: Disable. | 0 |

| | | | | |
|---|---|---|---|---|
| 4 | RB_UIE_FIFO_OV | RW | FIFO overflow interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 3 | RB_UIE_HST_SOF | RW | USB host mode, SOF receive completion interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 2 | RB_UIE_SUSPEND | RW | USB bus suspend or wakeup event interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 1 | RB_UIE_TRANSFER | RW | USB transfer completion interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 0 | RB_UIE_DETECT | RW | USB host mode, USB device connect or disconnect event interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| | RB_UIE_BUS_RST | RW | USB device mode, USB bus reset event interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |

#### 16.2.1.3 USB Device Address Register (R8_USB_DEV_AD)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved. | 0 |
| [6:0] | MASK_USB_ADDR [6:0] | RW | Host mode is the address or HUB address of the USB device currently in operation;<br>Device mode: the address of that USB itself. | 0 |

#### 16.2.1.4 USB Miscellaneous Status Register (R8_USB_MIS_ST)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UMS_SOF_PRES | RO | SOF packet presage status in USB host mode:<br>1: SOF packet is about to be transmitted, if there are other USB packets at this time, it will be automatically delayed;<br>0: No SOF packet is transmitted. | x |
| 6 | RB_UMS_SOF_ACT | RO | SOF packet transfer status in USB host mode:<br>1: SOF packet is being transmitted;<br>0: Transmit complete or idle. | x |
| 5 | RB_UMS_SIE_FREE | RO | USB protocol handler free:<br>1: The protocol device is free;<br>0: Busy, USB transfer in progress. | 1 |
| 4 | RB_UMS_R_FIFO_RDY | RO | USB receive FIFO data ready:<br>1: The receive FIFO is not empty;<br>0: The receive FIFO is empty. | 0 |
| 3 | RB_UMS_BUS_RST | RO | USB bus reset:<br>1: The current USB bus is in reset state;<br>0: The current USB bus is in a non-reset state. | x |
| 2 | RB_UMS_SUSPEND | RO | USB suspend:<br>1: The USB bus is in a suspended state, and there is no USB activity for a period of time;<br>0: The USB bus is in a non-suspend state. | 0 |
| 1 | RB_UMS_DM_LEVEL | RO | In USB host mode, level state of the DM pin when the device is just connected to the USB port is used to judge the speed:<br>1: High level/low-speed;<br>0: Low level/full-speed. | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | RB_UMS_DEV_ATTACH | RO | USB device attach status for the port in USB host mode:<br>1: The port has been connected to a USB device;<br>0: The port has no USB device connected. | 0 |

### 16.2.1.5 USB Interrupt Flag Register (R8_USB_INT_FG)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_U_IS_NAK | RO | NAK response interrupt flag bit in USB device mode:<br>1: Response to NAK during the current USB transfer;<br>0: No event. | 0 |
| 6 | RB_U_TOG_OK | RO | Current USB transmission DATA0/1 synchronization flag match status bit:<br>1: Synchronized; 0: Unsynchronized. | 0 |
| 5 | RB_U_SIE_FREE | RO | USB Protocol Handler free:<br>1: USB idle;<br>0: Busy, USB transfer in progress. | 1 |
| 4 | RB_UIF_FIFO_OV | RW | USB FIFO overflow interrupt flag, write 1 to clear:<br>1: FIFO overflow trigger;<br>0: No event. | 0 |
| 3 | RB_UIF_HST_SOF | RW | SOF timer interrupt flag in USB host mode, write 1 to clear:<br>1: SOF packet transmission completion trigger;<br>0: No event. | 0 |
| 2 | RB_UIF_SUSPEND | RW | USB bus suspended or wake-up event interrupt flag, write 1 to clear:<br>1: Triggered by USB suspend event or wake-up event;<br>0: No event. | 0 |
| 1 | RB_UIF_TRANSFER | RW | USB transfer completion interrupt flag, write 1 to clear:<br>1: A USB transfer is completed trigger;<br>0: No event. | 0 |
| 0 | RB_UIF_DETECT | RW | In USB host mode, USB device connect or disconnect event interrupt flag, write 1 to clear:<br>1: Detected USB device connection or disconnection trigger;<br>0: No event. | 0 |
| | RB_UIF_BUS_RST | RW | In USB device mode, USB bus reset event interrupt flag bit, write 1 to clear:<br>1: USB bus reset event trigger;<br>0: No event. | 0 |

### 16.2.1.6 USB Interrupt Status Register (R8_USB_INT_ST)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UIS_IS_NAK | RO | NAK response status bit in USB device mode, same as RB_U_IS_NAK:<br>1: Response to NAK during the current USB transfer;<br>0: No event. | 0 |

| | | | | |
|---|---|---|---|---|
| 6 | RB_UIS_TOG_OK | RO | Current USB transmission DATA0/1 synchronization flag match status bit:<br>1: Synchronized; 0: Unsynchronized. | 0 |
| [5:4] | MASK_UIS_TOKEN[1:0] | RO | In device mode, the token PID identifier of the current USB transfer transaction. | x |
| [3:0] | MASK_UIS_ENDP[3:0] | RO | In device mode, the endpoint number of the current USB transfer transaction. | x |
| | MASK_UIS_H_RES[3:0] | RO | In host mode, the response PID identifier of the current USB transmission transaction, 0000 means the device has no response or timed out; other values indicate the response PID. | x |

MASK_UIS_TOKEN is used in USB device mode to identify the token PID of the current USB transmission transaction: 00 for OUT packets; 01 reserved; 10 for IN packets; 11 for SETUP packets.
MASK_UIS_H_RES is only valid in host mode. In host mode, if the host sends an OUT/SETUP token packet, this PID is either a handshake packet ACK/NAK/STALL, or the device has no answer/timeout. If the host sends an IN token packet, this PID is either the packet PID (DATA0/DATA1) or the handshake packet PID.

#### 16.2.1.7 USB Receive Length Register (R16_USB_RX_LEN)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:10] | Reserved | RO | Reserved. | 0 |
| [9:0] | R16_USB_RX_LEN[9:0] | RO | The current number of data bytes received by the USB endpoint | X |

### 16.2.2 USB Device Register Description

In USB device mode, USBFS module provides 8 groups of bidirectional endpoints from endpoints 0-7. The maximum packet length of all endpoints except endpoint 3 is 64 bytes, and the maximum packet length of endpoint 3 is 1023 bytes.

- Endpoint 0 is the default endpoint, which supports control transmission and shares a 64-byte data buffer between transmit and receive.
- Endpoints 1-7 each include a transmit endpoint IN and a receive endpoint OUT, each with a separate data buffer for transmit and receive, and support bulk, interrupt and real-time/synchronous transfers.
- Endpoint 0 has a separate DMA address, shared between transmit and receive, and endpoints 1-7 each have a DMA address for transmit and receive. The data buffer mode can be set to double buffer or single buffer by entering the RB_UEPn_BUF_MOD register. If the double buffer mode is used, the endpoint can only use single direction transmission.
- Each group of endpoints has a transceiver control register R16_UEPn_CTRL and a transmission length register R16_UEPn_T_LEN and R32_UEPN_DMA (n=0~7), which are used to configure the synchronization trigger bit, the response to the OUT transaction and the IN transaction, and the length of the transmitted data.

As a necessary USB device, the pull-up resistor of USB bus can be set by software at any time. When RB_UC_DEV_PU_EN in the USB control register R8_USB_CTRL is set to 1, the controller connects the pull-up resistor to the DP/DM pin of USB bus internally according to the speed setting of RB_UC_SPEED_TYPE, and enables the USB device function.

When USB bus reset, USB bus suspension or wake-up events are detected, or when USB successfully processes data transmission or data reception, the USB protocol processor will set corresponding interrupt flags, and if

interrupt enable is turned on, it will also generate corresponding interrupt requests. The application program can query and analyze the interrupt flag register R8_USB_INT_FG directly or in the USB interrupt service program, and perform corresponding processing according to RB_UIF_BUS_RST and RB_UIF_SUSPEND; Moreover, if RB_UIF_TRANSFER is valid, it is necessary to continue to analyze the USB interrupt status register R8_USB_INT_ST, and make corresponding processing according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identifier MASK_UIS_TOKEN. If the synchronization trigger bit RB_UEP_R_TOG of the OUT transaction of each endpoint is set in advance, it can be judged whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the endpoint by RB_U_TOG_OK or RB_UIS_TOG_OK, and if the data is synchronized, the data is valid; If the data is not synchronized, the data should be discarded. After processing the USB transmission or reception interrupt every time, the synchronization trigger bit of the corresponding endpoint should be correctly modified for detecting whether the next transmitted data packet or the next received data packet is synchronized; In addition, setting RB_UEP_T_AUTO_TOG or RB_UEP_R_AUTO_TOG can automatically modify the corresponding synchronization trigger bit (flip or auto-decrement) after successful transmission or reception.

The data to be sent by each endpoint is in its own buffer, and the length of the data to be sent is independently set in R16_UEPn_T_LEN; The data received by each endpoint is in its own buffer, but the length of the received data is in the USB reception length register R16_USB_RX_LEN, which can be distinguished according to the current endpoint number when USB reception is interrupted.

Table 16-3 Device-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_UDEV_CTRL | 0x40023401 | USB device physical port control register | 0xX0 |
| R8_UEP4_1_MOD | 0x4002340C | Endpoint 1/4 mode control register | 0x00 |
| R8_UEP2_3_MOD | 0x4002340D | Endpoint 2/3 mode control register | 0x00 |
| R8_UEP5_6_MOD | 0x4002340E | Endpoint 5/6 mode control register | 0x00 |
| R8_UEP7_MOD | 0x4002340F | Endpoint 7 mode control register | 0x00 |
| R32_UEP0_DMA | 0x40023410 | Endpoint 0 buffer start address | 0x0000XXXX |
| R32_UEP1_DMA | 0x40023414 | Endpoint 1 buffer start address | 0x0000XXXX |
| R32_UEP2_DMA | 0x40023418 | Endpoint 2 buffer start address | 0x0000XXXX |
| R32_UEP3_DMA | 0x4002341C | Endpoint 3 buffer start address | 0x0000XXXX |
| R32_UEP4_DMA | 0x40023420 | Endpoint 4 buffer start address | 0x0000XXXX |
| R32_UEP5_DMA | 0x40023424 | Endpoint 5 buffer start address | 0x0000XXXX |
| R32_UEP6_DMA | 0x40023428 | Endpoint 6 buffer start address | 0x0000XXXX |
| R32_UEP7_DMA | 0x4002342C | Endpoint 7 buffer start address | 0x0000XXXX |
| R32_USB_EP0_CTRL | 0x40023430 | Endpoint 0 transmit length and control registers | 0x000000XX |
| R16_UEP0_T_LEN | 0x40023430 | Endpoint 0 transmit length register | 0x00XX |
| R16_UEP0_CTRL | 0x40023432 | Endpoint 0 control register | 0x0000 |
| R32_USB_EP1_CTRL | 0x40023434 | Endpoint 1 transmit length and control registers | 0x000000XX |
| R16_UEP1_T_LEN | 0x40023434 | Endpoint 1 transmit length register | 0x00XX |
| R16_UEP1_CTRL | 0x40023436 | Endpoint 1 control register | 0x0000 |
| R32_USB_EP2_CTRL | 0x40023438 | Endpoint 2 transmit length and control registers | 0x000000XX |
| R16_UEP2_T_LEN | 0x40023438 | Endpoint 2 transmit length register | 0x00XX |
| R16_UEP2_CTRL | 0x4002343A | Endpoint 2 control register | 0x0000 |
| R32_USB_EP3_CTRL | 0x4002343C | Endpoint 3 transmit length and control registers | 0x00000XXX |

| | | | |
|---|---|---|---|
| R16_UEP3_T_LEN | 0x4002343C | Endpoint 3 transmit length register | 0x0XXX |
| R16_UEP3_CTRL | 0x4002343E | Endpoint 3 control register | 0x0000 |
| R32_USB_EP4_CTRL | 0x40023440 | Endpoint 4 transmit length and control registers | 0x000000XX |
| R16_UEP4_T_LEN | 0x40023440 | Endpoint 4 transmit length register | 0x00XX |
| R16_UEP4_CTRL | 0x40023442 | Endpoint 4 control register | 0x0000 |
| R32_USB_EP5_CTRL | 0x40023444 | Endpoint 5 transmit length and control registers | 0x000000XX |
| R16_UEP5_T_LEN | 0x40023444 | Endpoint 5 transmit length register | 0x00XX |
| R16_UEP5_CTRL | 0x40023446 | Endpoint 5 control register | 0x0000 |
| R32_USB_EP6_CTRL | 0x40023448 | Endpoint 6 transmit length and control registers | 0x000000XX |
| R16_UEP6_T_LEN | 0x40023448 | Endpoint 6 transmit length register | 0x00XX |
| R16_UEP6_CTRL | 0x4002344A | Endpoint 6 control register | 0x0000 |
| R32_USB_EP7_CTRL | 0x4002344C | Endpoint 7 transmit length and control registers | 0x000000XX |
| R16_UEP7_T_LEN | 0x4002344C | Endpoint 7 transmit length register | 0x00XX |
| R16_UEP7_CTRL | 0x4002344E | Endpoint 7 control register | 0x0000 |

#### 16.2.2.1 USB Device Physical Port Control Register (R8_UDEV_CTRL)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UD_PD_DIS | RW | USB device port UDP/UDM pin internal pull-down resistor control bits:<br>1: Disable the internal pull-down;<br>0: Enable internal pull-down, also used in GPIO mode to provide pull-down resistor.<br>*Note: MODE and CNF of GPIOC_CFGXR and GPIOC_OUTDR have been changed to set the pull-down, this bit is reserved.* | 1 |
| 6 | Reserved | RO | Reserved. | 0 |
| 5 | RB_UD_DP_PIN | RO | Current UDP pin status:<br>1: High;<br>0: low. | X |
| 4 | RB_UD_DM_PIN | RO | Current UDM pin state:<br>1: High;<br>0: low. | X |
| 3 | Reserved | RO | Reserved. | 0 |
| 2 | RB_UD_LOW_SPEED | RW | USB device physical port low speed mode enable bit:<br>1: Select 1.5Mbps low speed mode;<br>0: Select 12Mbps full speed mode. | 0 |
| 1 | RB_UD_GP_BIT | RW | USB device mode general flag bit, user-defined. | 0 |
| 0 | RB_UD_PORT_EN | RW | USB device physical port enable bit:<br>1: Enable the physical port;<br>0: Disable the physical port. | 0 |

#### 16.2.2.2 Endpoint 1/4 Mode Control Registers (R8_UEP4_1_MOD)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UEP1_RX_EN | RW | 1: Enable endpoint 1 receive (OUT);<br>0: Disable endpoint 1 receive. | 0 |
| 6 | RB_UEP1_TX_EN | RW | 1: Enable endpoint 1 transmit (IN);<br>0: Disable endpoint 1 transmit. | 0 |
| 5 | Reserved | RO | Reserved | 0 |

| | | | | |
|---|---|---|---|---|
| 4 | RB_UEP1_BUF_MOD | RW | Endpoint 1 data buffer mode control bit. | 0 |
| 3 | RB_UEP4_RX_EN | RW | 1: Enable endpoint 4 receive (OUT);<br>0: Disable endpoint 4 receive. | 0 |
| 2 | RB_UEP4_TX_EN | RW | 1: Enable endpoint 4 transmit (IN);<br>0: Disable endpoint 4 transmit. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_UEP4_BUF_MOD | RW | Endpoint 4 data buffer mode control bit. | 0 |

#### 16.2.2.3 Endpoint 2/3 Mode Control Registers (R8_UEP2_3_MOD)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UEP3_RX_EN | RW | 1: Enable endpoint 3 receive (OUT);<br>0: Disable endpoint 3 receive. | 0 |
| 6 | RB_UEP3_TX_EN | RW | 1: Enable endpoint 3 transmit (IN);<br>0: Disable endpoint 3 transmit. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UEP3_BUF_MOD | RW | Endpoint 3 data buffer mode control bit. | 0 |
| 3 | RB_UEP2_RX_EN | RW | 1: Enable endpoint 2 receive (OUT);<br>0: Disable endpoint 2 receive. | 0 |
| 2 | RB_UEP2_TX_EN | RW | 1: Enable endpoint 2 transmit (IN);<br>0: Disable endpoint 2 transmit. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_UEP2_BUF_MOD | RW | Endpoint 2 data buffer mode control bit. | 0 |

#### 16.2.2.4 Endpoint 5/6 Mode Control Registers (R8_UEP5_6_MOD)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UEP6_RX_EN | RW | 1: Enable endpoint 6 receive (OUT);<br>0: Disable endpoint 6 receive. | 0 |
| 6 | RB_UEP6_TX_EN | RW | 1: Enable endpoint 6 transmit (IN);<br>0: Disable endpoint 6 transmit. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UEP6_BUF_MOD | RW | Endpoint 6 data buffer mode control bit. | 0 |
| 3 | RB_UEP5_RX_EN | RW | 1: Enable endpoint 5 receive (OUT);<br>0: Disable endpoint 5 receive. | 0 |
| 2 | RB_UEP5_TX_EN | RW | 1: Enable endpoint 5 transmit (IN);<br>0: Disable endpoint 5 transmit. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_UEP5_BUF_MOD | RW | Endpoint 5 data buffer mode control bit. | 0 |

#### 16.2.2.5 Endpoint 7 Mode Control Registers (R8_UEP7_MOD)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | Reserved | RO | Reserved | 0 |
| 3 | RB_UEP7_RX_EN | RW | 1: Enable endpoint 7 receive (OUT);<br>0: Disable endpoint 7 receive. | 0 |
| 2 | RB_UEP7_TX_EN | RW | 1: Enable endpoint 7 transmit (IN);<br>0: Disable endpoint 7 transmit. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_UEP7_BUF_MOD | RW | Endpoint 7 data buffer mode control bit. | 0 |

The data buffer modes of USB endpoints 1-7 are configured by the combination of RB_UEPn_RX_EN, RB_UEPn_TX_EN and RB_UEPn_BUF_MOD respectively. See Table 16-4 for details. Among them, in the double 64-byte buffer mode, the first 64-byte buffer will be selected according to RB_UEP_*_TOG=0, and the last 64-byte buffer will be selected according to RB_UEP_*_TOG=1. Setting Rb _ UEP _ * _ TOG = 1 can realize automatic

switching.

Table 16-4 Endpoint N Buffer Mode (n=1-7)

| RB_UEPn_RX_EN | RB_UEPn_TX_EN | RB_UEPn_BUF_MOD | Description: The address starting with R16_UEPn_DMA is arranged from low to high. |
|---|---|---|---|
| 0 | 0 | X | Endpoint is disabled and R16_UEPn_DMA buffer is not used. |
| 1 | 0 | 0 | Single 64-byte receive buffer (OUT). |
| 1 | 0 | 1 | Double 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG. |
| 0 | 1 | 0 | SINgle 64-byte transmit buffer (IN). |
| 0 | 1 | 1 | Double 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG. |
| 1 | 1 | 0 | Single 64-byte receive buffer (OUT) and single 64-byte transmit buffer (IN). |
| 1 | 1 | 1 | Double 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG,<br>Double 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG.<br>All 256 bytes are arranged as follows:<br>UEPn_DMA+0 address: The endpoint receive address when RB_UEP_R_TOG=0;<br>UEPn_DMA+64 address: The endpoint receive address when RB_UEP_R_TOG=1;<br>UEPn_DMA+128 address: The endpoint transmit address when RB_UEP_T_TOG=0;<br>UEPn_DMA+192 address: The endpoint transmit address when RB_UEP_T_TOG=1. |

### 16.2.2.6 Endpoint n Buffer Start Address (R32_UEPn_DMA) (n=0-7)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:14] | Reserved | RO | Reserved | 0 |
| [13:0] | UEPn_DMA[13:0] | RW | Start address of endpoint n buffer.<br>The lower 14 bits are valid, and the address must be 4 bytes aligned. | x |

### 16.2.2.7 Endpoint n Transmit Length and Control Registers (R32_USB_EPn_CTRL) (n=0-7)

| Bit | Name |
|---|---|
| [31:16] | R16_UEPn_CTRL |
| [15:0] | R16_UEPn_T_LEN |

### 16.2.2.8 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=0-1)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:7] | Reserved | RO | Reserved. | 0 |
| [6:0] | R8_UEPn_T_LEN[6:0] | RW | Set the number of data bytes n=0, 1 that USB endpoint n is ready to send. | x |

### 16.2.2.9 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=2)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved. | 0 |
| 7 | HOST_PID3 | RW | PID[3] in host mode. | 0 |

| | | | | |
|---|---|---|---|---|
| [6:0] | R8_UEPn_T_LEN[6:0] | RW | Set the number of data bytes n=2 that USB endpoint n is ready to send. | x |

#### 16.2.2.10 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=3)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:10] | Reserved | RO | Reserved. | 0 |
| [9:0] | R8_UEPn_T_LEN[9:0] | RW | Set the number of data bytes n=3 that USB endpoint n is ready to send. | x |

#### 16.2.2.11 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=4-7)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:10] | Reserved | RO | Reserved. | 0 |
| [9:0] | R8_UEPn_T_LEN[6:0] | RW | Set the number of data bytes n=4, 5, 6, 7 that USB endpoint n is ready to send. | x |

#### 16.2.2.12 Endpoint n Control Register (R16_UEPn_CTRL) (n=0-7)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | Reserved | RO | Reserved | 0 |
| 11 | RB_UEP_R_AUTO_TOG | RW | Synchronization trigger bit auto-flip enable control bit:<br>1: Automatically flip the corresponding synchronization trigger bit after successful data reception;<br>0: It does not turn over automatically, and can be switched manually.<br>*Note: This bit of endpoint 0 is reserved.* | 0 |
| 10 | MASK_UEP_R_TOG | RW | Expected synchronization trigger bit of the receiver of USB endpoint n (processing the OUT transaction):<br>1: Expected DATA1; 0: Expected DATA0。 | 0 |
| [9:8] | MASK_UEP_R_RES[1:0] | RW | Response control of the receiver of endpoint n to OUT transaction:<br>00: Answer ACK;<br>01: Timeout/no response, used for real-time/synchronous transmission of non-endpoint 0;<br>10: Answer NAK or busy;<br>11: Answer STALL or error. | 0 |
| [7:4] | Reserved | RO | Reserved | 0 |
| 3 | RB_UEP_T_AUTO_TOG | RW | Synchronization trigger bit auto-flip enable control bit:<br>1: Automatically flip the corresponding synchronization trigger bit after successful data transmission;<br>0: It does not turn over automatically, and can be switched manually.<br>*Note: This bit of endpoint 0 is reserved.* | 0 |
| 2 | RB_UEP_T_TOG | RW | Synchronization trigger bits prepared by the transmitter (processing in transaction) of USB endpoint N:<br>1: Transmit DATA1； ; 0: Transmit DATA0. | 0 |
| [1:0] | MASK_UEP_T_RES[1:0] | RW | Response control of the receiver of endpoint n to IN transaction：<br>00: DATA0/DATA1 data is ready and an ACK | 0 |

| | | | | |
|---|---|---|---|---|
| | | | is expected;<br>01: Answer DATA0/DATA1 and expect no response, which is used for real-time/synchronous transmission of non-endpoint 0;<br>10: Answer NAK or busy;<br>11: Answer STALL or error. | |

### 16.2.3 USB Host Register Description

In USB host mode, the chip provides a set of bi-directional host endpoints, including a transmit endpoint OUT and a receive endpoint IN. The maximum length of a packet is 1024 bytes (Simultaneous transfers), supporting control transfers, interrupt transfers, bulk transfers and real-time/synchronous transfers.

The RB_UIF_TRANSFER interrupt flag is always automatically set at the end of processing for each USB transaction initiated by the host endpoint. The application can query directly or in the USB interrupt service program and analyze the interrupt flag register R8_USB_INT_FG to process each interrupt flag accordingly; and, if RB_UIF_TRANSFER is valid, then it also needs to continue to analyze the USB interrupt status register R8_USB_INT_ST, based on the current USB transfer If RB_UIF_TRANSFER is valid, then the USB interrupt status register R8_USB_INT_ST should be analyzed further and processed accordingly according to the current USB transfer transaction's answer PID identification MASK_UIS_H_RES.

If the sync trigger bit (RB_UH_R_TOG) of the IN transaction of the host receive endpoint is set in advance, then it can be judged by RB_U_TOG_OK or RB_UIS_TOG_OK whether the sync trigger bit of the currently received packet matches the sync trigger bit of the host receive endpoint, if the data is synchronous, then the data is valid; if the data is not synchronous, then the data should be discarded. After each USB transmit or receive interrupt is processed, the synchronization trigger bit of the corresponding host endpoint should be correctly modified to synchronize the next transmitted packet and to check whether the next received packet is synchronized; in addition, by setting RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG, the corresponding synchronization trigger bit can be automatically flipped after a successful transmit or receive. synchronization trigger bits.

The USB host token setting register R8_UH_EP_PID is used to set the endpoint number of the operated target device and the token PID package identification of this USB transmission transaction. The data corresponding to the SETUP token and the OUT token are provided by the host sending endpoint, the data to be sent is in R16_UH_TX_DMA buffer, and the length of the data to be sent is set in R16_UH_TX_LEN; The data corresponding to the in token is returned by the target device to the host receiving endpoint, and the received data is stored in R16_UH_RX_DMA buffer, and the length of the received data is stored in R16_USB_RX_LEN.

Table 16-5 Host-related registers list

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_UHOST_CTRL | 0x40023401 | USB host physical port control register | 0xX0 |
| R8_UH_EP_MOD | 0x4002340D | USB host endpoint mode control register | 0x00 |
| R16_UH_RX_DMA | 0x40023418 | USB host receive buffer start address | 0xXXXX |
| R16_UH_TX_DMA | 0x4002341C | USB host transmit buffer start address | 0xXXXX |
| R16_UH_SETUP | 0x40023436 | USB host auxiliary setup register | 0x0000 |
| R8_UH_EP_PID | 0x40023438 | USB host token setup register | 0x00 |
| R8_UH_RX_CTRL | 0x4002343B | USB host receive endpoint control register | 0x00 |
| R16_UH_TX_LEN | 0x4002343C | USB host transmit length register | 0xXXXX |
| R8_UH_TX_CTRL | 0x4002343E | USB host transmit endpoint control register | 0x00 |

#### 16.2.3.1 USB Host Control Register (R8_UHOST_CTRL)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UH_PD_DIS | RW | USB host port UD+/UD- pin internal pull-down resistor control bits:<br>1: Disable the internal pull-down;<br>0: Enable internal pull-down.<br>Also used in GPIO mode to provide pull-down resistor. | 1 |
| 6 | Reserved | RO | Reserved. | 0 |
| 5 | RB_UH_DP_PIN | RO | Current UD+ pin status:<br>1: High level;<br>0: Low level. | x |
| 4 | RB_UH_DM_PIN | RO | Current UD- pin state:<br>1: High;<br>0: Low. | x |
| 3 | Reserved | RO | Reserved. | 0 |
| 2 | RB_UH_LOW_SPEED | RW | USB host port low-speed mode enable bit:<br>1: Select 1.5Mbps low-speed mode;<br>0: Select 12Mbps full-speed mode. | 0 |
| 1 | RB_UH_BUS_RESET | RW | USB host mode bus reset control bit:<br>1: Force output USB bus reset;<br>0: End output. | 0 |
| 0 | RB_UH_PORT_EN | RW | USB host port enable bit:<br>1: Enable the host port;<br>0: Disable the host port.<br>This is automatically cleared to 0 when the USB device is disconnected. | 0 |

#### 16.2.3.2 USB Host Endpoint Mode Control Register (R8_UH_EP_MOD)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved. | 0 |
| 6 | RB_UH_EP_TX_EN | RW | Host sends endpoint transmit (SETUP/OUT) enable bits:<br>1: Enable endpoint transmission;<br>0: Disable endpoint transmission. | 0 |
| 5 | Reserved | RO | Reserved. | 0 |
| 4 | RB_UH_EP_TBUF_MOD | RW | Host transmit endpoint transmit data buffer mode control bit. | 0 |
| 3 | RB_UH_EP_RX_EN | RW | Host receive endpoint receive (IN) enable bit:<br>1: Enable endpoint reception;<br>0: Disable endpoint reception. | 0 |
| [2:1] | Reserved | RO | Reserved. | 00b |
| 0 | RB_UH_EP_RBUF_MOD | RW | USB host receive endpoint receive data buffer mode control bit. | 0 |

The host transmit endpoint data buffer mode is controlled by a combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD, refer to the table below.

Table 16-6 Host transmit buffer mode

| RB_UH_EP_TX_EN | RB_UH_EP_TBUF_MOD | Description: R16_UH_TX_DMA as starting address |
|---|---|---|
| 0 | X | Endpoint is disabled and the R16_UH_TX_DMA buffer is not used. |
| 1 | 0 | Single 64-byte transmit buffer (SETUP/OUT) |
| 1 | 1 | Double 64-byte transmit buffer, selected via RB_UH_T_TOG:<br>Selects the first 64-byte buffer when RB_UH_T_TOG=0; |

| | | |
|---|---|---|
| | | Selects the back 64-byte buffer when RB_UH_T_TOG=1. |

The host receive endpoint data buffer mode is controlled by the combination of RB_UH_EP_RX_EN and RB_UH_EP_RBUF_MOD, refer to the table below.

Table 16-7 Host receive buffer mode

| RB_UH_EP_RX_EN | RB_UH_EP_RBUF_MOD | Structure description: R16_UH_TX_DMA as starting address |
|---|---|---|
| 0 | X | Endpoint is disabled and the R16_UH_RX_DMA buffer is not used. |
| 1 | 0 | Single 64-byte receive buffer (IN). |
| 1 | 1 | Double 64-byte receive buffer, selected via RB_UH_R_TOG:<br>Selects the first 64-byte buffer when RB_UH_R_TOG=0;<br>Selects the back 64-byte buffer when RB_UH_R_TOG=1. |

#### 16.2.3.3 USB Host Receive Buffer Start Address (R16_UH_RX_DMA)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_UH_RX_DMA[15:0] | RW | Start address of the host endpoint data receive buffer.<br>The lower 15 bits are valid and the address must be 4-byte aligned. | X |

#### 16.2.3.4 USB Host Transmit Buffer Start Address (R16_UH_TX_DMA)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_UH_TX_DMA[15:0] | RW | Start address of the host endpoint data transmit buffer.<br>The lower 15 bits are valid and the address must be 4-byte aligned. | X |

#### 16.2.3.5 USB Host Auxiliary Setup Register (R16_UH_SETUP)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:11] | Reserved | RO | Reserved | 0 |
| 10 | RB_UH_PRE_PID_EN | RW | Low-speed leading packet PRE PID enable bit:<br>1: Enabled for communication with low-speed USB devices via the external HUB.<br>0: Disable the low-speed preamble. | 0 |
| [9:3] | Reserved | RO | Reserved | 0 |
| 2 | RB_UH_SOF_EN | RW | Automatic SOF packet generation enable bit:<br>1: The host automatically generates SOF packets;<br>0: Not automatically generated, but can be generated manually. | 0 |
| [1:0] | Reserved | RO | Reserved | 0 |

#### 16.2.3.6 USB Host Token Setup Register (R8_UH_EP_PID)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | MASK_UH_TOKEN[3:0] | RW | Set the token PID packet identifier for this USB transfer transaction. | 0 |
| [3:0] | MASK_UH_ENDP[3: | RW | Set the endpoint number of the target | 0 |

| | | | | |
|---|---|---|---|---|
| | 0] | | device being operated on this time. | |

### 16.2.3.7 USB Host Receive Endpoint Control Register (R8_UH_RX_CTRL)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UH_R_TOG | RW | Desired synchronization trigger bits for USB host receivers (Handling IN transactions):<br>1: Expect DATA1;<br>0: Expect DATA0. | 0 |
| [6:5] | Reserved | RO | Reserved. | 00b |
| 4 | RB_UH_R_AUTO_TOG | RW | Sync trigger bit auto-flip enable control bit:<br>1: Automatically flips the corresponding desired sync trigger bit (RB_UH_R_TOG) upon successful data reception;<br>0: Not automatically flipped, can be manually toggled. | 0 |
| 3 | Reserved | RO | Reserved. | 0 |
| 2 | RB_UH_R_RES | RW | Control bits for host receiver response to IN transactions:<br>1: No response, for real-time/synchronous transmission of non-zero endpoints;<br>0: Answer to ACK. | 0 |
| [1:0] | Reserved | RO | Reserved. | 00b |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | Reserved | RO | Reserved | 0 |
| 3 | RB_UH_R_AUTO_TOG | RW | Sync trigger bit auto-flip enable control bit:<br>1: Automatically flips the corresponding desired sync trigger bit (RB_UH_R_TOG) upon successful data reception;<br>0: Not automatically flipped, can be manually toggled. | 0 |
| 2 | RB_UH_R_TOG | RW | Expected synchronization trigger bits of USB host receiver (processing in transaction):<br>1: Expected DATA1;<br>0: Expected DATA0. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_UH_R_RES | RW | Host receiver response control bits for IN transactions:<br>1: No response for real-time/synchronous transmission of non-0 endpoints;<br>0: Response to ACK. | 0 |

### 16.2.3.8 USB Host Transmit Length Register (R16_UH_TX_LEN)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R8_UH_TX_LEN[15:0] | RW | Sets the number of bytes of data that the USB host sending endpoint is ready to transmit. | X |

### 16.2.3.9 USB Host Transmit Endpoint Control Register (R8_UH_TX_CTRL)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | Reserved | RO | Reserved. | 0 |

| | | | | |
|---|---|---|---|---|
| 3 | RB_UH_T_AUTO_TOG | RW | Synchronous trigger bit auto-flip enable control bit:<br>1: The corresponding synchronous trigger bit (RB_UH_T_TOG) is automatically flipped upon successful data transmission;<br>0: Not automatically flipped, can be switched manually. | 0 |
| 2 | RB_UH_T_TOG | RW | Sync trigger bits prepared by the USB host sender (which handles SETUP/OUT transactions):<br>1: Indicates that DATA1 is sent;<br>0: Indicates that DATA0 is sent. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_UH_T_RES | RW | USB host transmitter response control bits for SETUP/OUT transactions:<br>1: No response expected, for real-time/synchronous transmission of non-0 endpoints;<br>0: Expect an answer ACK. | 0 |

# Chapter 17 Operational Amplifier (OPA) and Comparator (CMP)

The module includes four OPA operational amplifiers (OPA1, OPA2, OPA3 and OPA4) and three CMP voltage comparators (CMP1, CMP2 and CMP3). These OPA and CMP can be combined into two groups of AC small signal amplifiers and decoders (QII1 and QII2) and 2 groups of differential input current sampling (ISP1 and ISP2).

## 17.1 Main Features

Main features of OPA and CMP:

- OPA1/OPA2 support self-biased programmable gain operational amplifier (PGA).
- OPA3/OPA4 support single-ended and differential inputs, and support PGA magnification selection.
- The output channels of OPA1/OPA2/OPA3/OPA4 are directly connected to the internal channel of ADC or the input of comparator.
- CMP1/CMP2/CMP3 support optional digital filtering and hysteresis.
- The negative input of CMP2/CMP3 supports optional internal self-bias voltage.
- 1 interrupt vector

Main features of OPA and CMP combined application:

- The QII1/QII2 output channel is directly connected to the ADC internal channel or the comparator input.
- QII1/QII2 supports PGA self-bias and gain selection, and supports digital filtering.
- The negative input of the comparator (CMP2) in QII2 supports an optional internal self-bias voltage.
- ISP1/ISP2 supports differential or single-ended selection, and the gain and output DC level can be configured.
- ISP1/ISP2 output channel is directly connected to ADC internal channel or comparator input.

## 17.2 Function Description

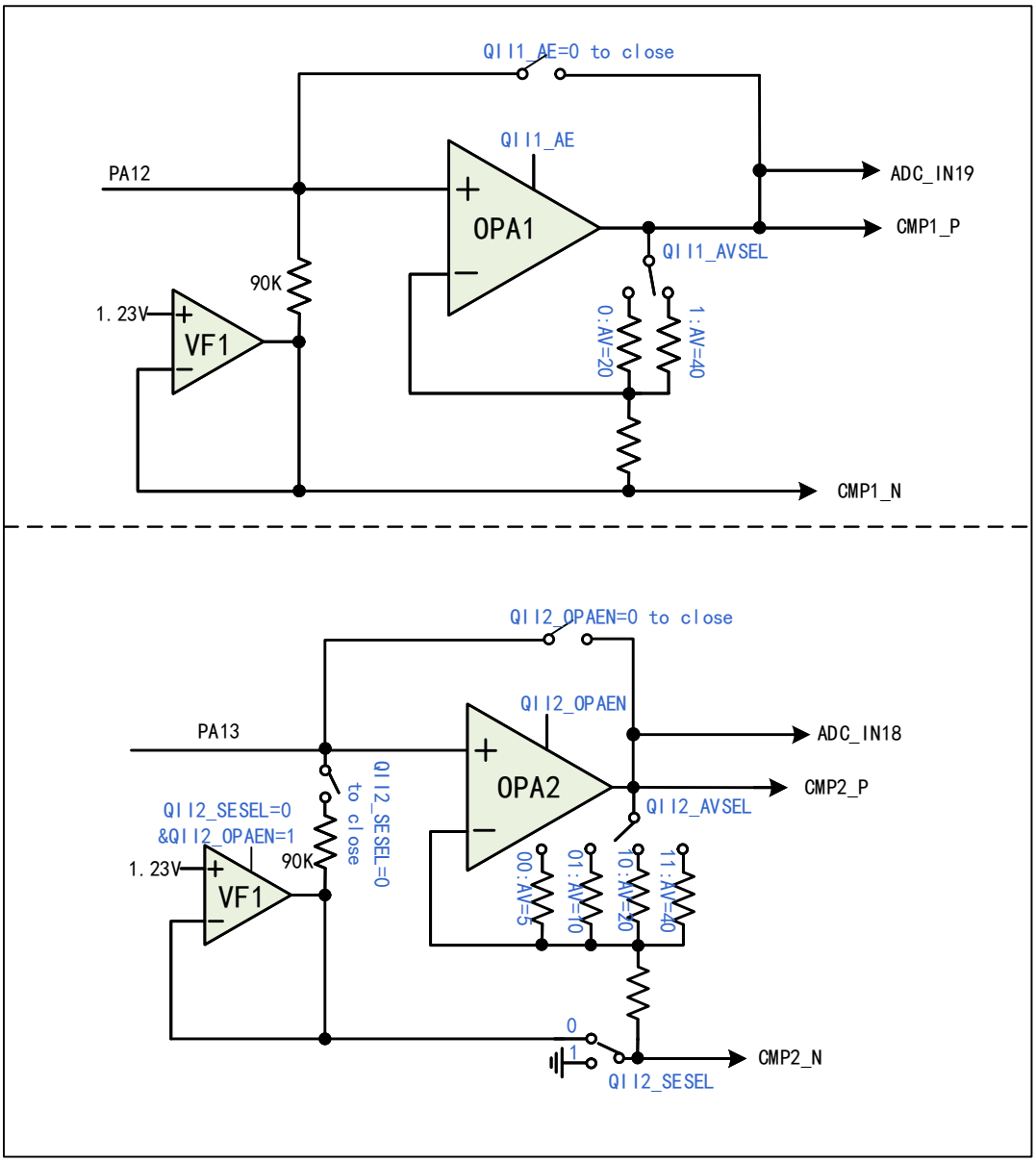
### 17.2.1 OPA1/OPA2

OPA1 and OPA2 support self-biased programmable gain op-amps (PGAs). Among them, the output result of OPA1 is connected to voltage comparator CMP1 and ADC channel IN19 inside the chip; the output result of OPA2 is connected to voltage comparator CMP2 and ADC channel IN18 inside the chip.

In the QII_CFGR register: configure the QII1_AE bit to enable OPA1 and VF1; configure the QII1_AVSEL bit to select the OPA1 op amp gain size.

In QII_CFGR register: configure QII2_OPAEN bit to enable OPA2; configure QII2_AVSEL bit to select OPA2 op-amp gain size; configure QII2_SESEL bit to select op-amp operating mode.

Figure 17-1 OPA1 and OPA2 structure diagram

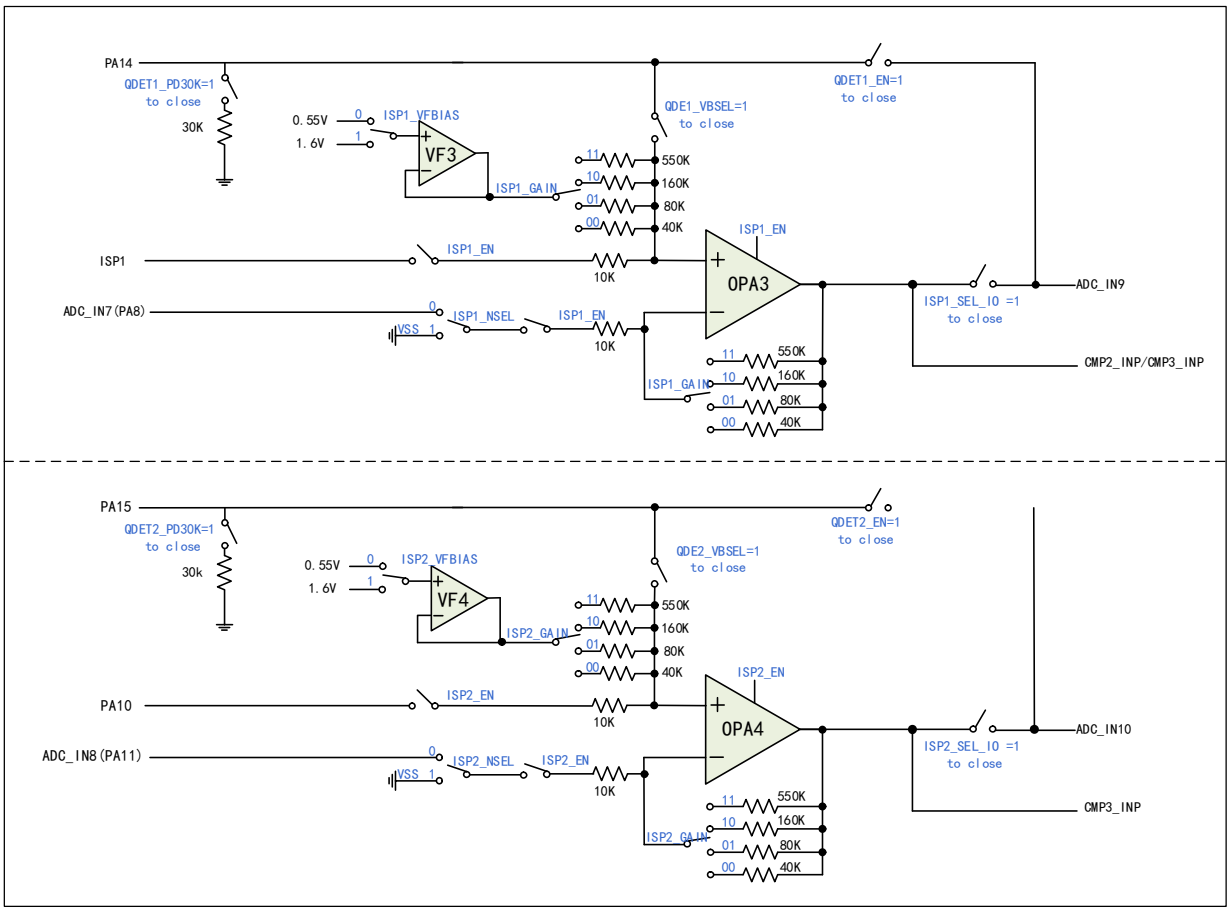



### 17.2.2 OPA3/OPA4

The OPA3 and OPA4 support single-ended and differential inputs, with PGA amplification selectable by changing the configuration, and also provide internal self-bias voltage. The outputs of OPA3 are internally connected to voltage comparator CMP2 or CMP3 and ADC channel IN9, while the outputs of OPA4 are internally connected to voltage comparator CMP3 and ADC channel IN10.

In the ISP_CTLR register: configure the ISPx_EN bit to enable the OPA module; Configure the ISPx_GAIN[1:0] bits to enable the magnitude of the operational amplifier gain; Configure the ISPx_VFBIAS bit and select the module reference voltage; Configure the ISPx_NSEL bit and the negative input of the module; Configure the ISPx_SEL_IO bit to enable the module to output to ADC; Configure QDETx_EN bit to enable q value detection; Configure the DETx_PD30K bit to enable the Q value detection 30K resistor; Configure the QDETx_VBSEL bit to enable the q-value detection reference voltage.

Figure 17-2 OPA3 and OPA4 structure diagram




### 17.2.3 CMP1/CMP2

CMP1 supports optional hysteresis characteristics, digital filtering at the output, and optional output filtering.

CMP2 supports optional hysteresis, optional internal N-terminal bias and digital filtering at the output. Its P-side channel can be input by GPIO or connected with OPA in the chip. Its N-terminal channel can be input by GPIO or connected with DAC output inside the chip.

In QII_CFGR register: configure QII1_AE bit to enable CMP1 module; Configure the QII1_HYPSEL bit to select CMP hysteresis voltage.

In the CMP_CTLR register: configure the CMP1_FILT_EN bit to enable the CMP1 output digital filtering function; configure the CMP1_FILT_CFG[4:0] bits to select the CMP1 output digital sampling interval, with the sampling interval set to be greater than the noise width and less than twice the noise width; configure the CMP1_TO_IO bits to enable the CMP1 output to the IO.

In the CMP_STATR register: the CMP1_OUT/CMP1_OUT_FILT bits allow you to read the output of the CMP1 filter function off/on.

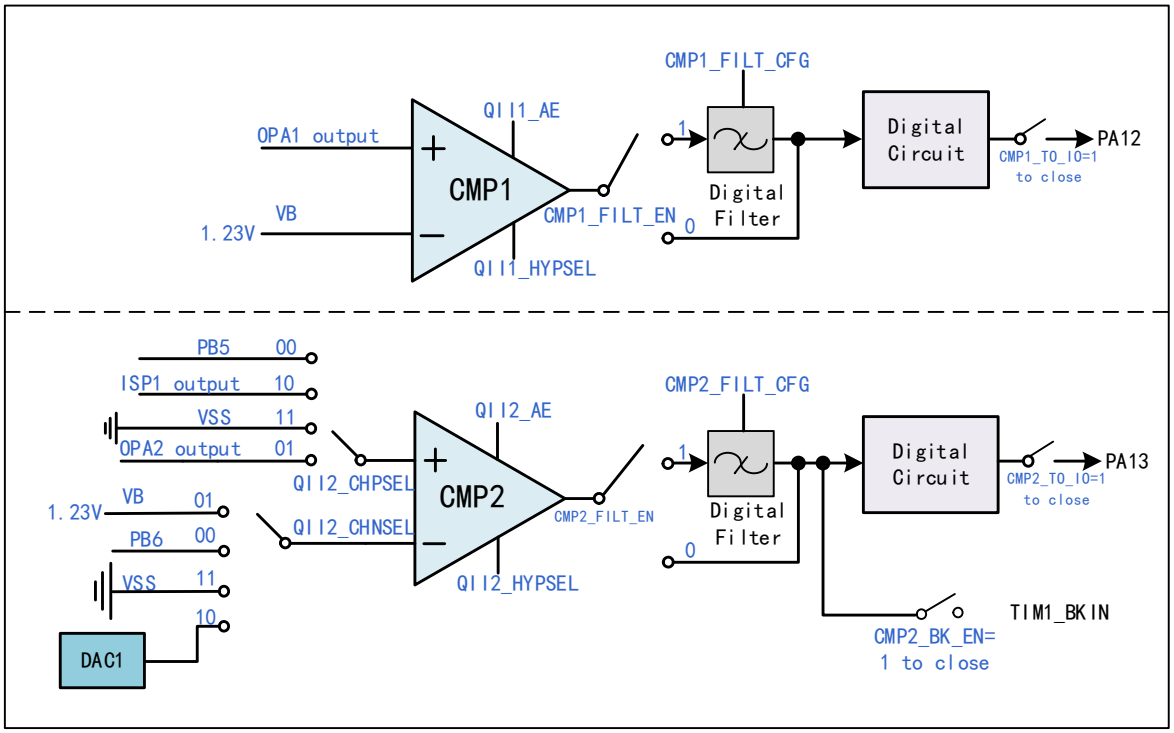
In the QII_CFGR register: configure the QII2_AE bit to enable the CMP2 module; configure the QII2_HYPSEL bit to select the comparator hysteresis voltage.

In the CMP_CTLR register: configure the CMP2_FILT_EN bit to enable the CMP2 output digital filtering function; configure the CMP2_FILT_CFG[4:0] bits to select the CMP2 output digital sampling interval, and the sampling

interval setting should be greater than the noise width and less than twice the noise width; configure the CMP2_TO_IO bit to enable the CMP2 output internal connection to IO function; configure CMP2_BK_EN bit to enable CMP2 output to be connected to Timer1 brake input; CMP2_INT_EN bit can be configured to enable CMP2 output interrupt.

In the CMP_STATR register: with the CMP2_OUT/CMP2_OUT_FILT bits, the output result of the CMP2 filtering function off/on can be read; with the CMP2_COMIF bit, the CMP2 output for high level interrupt flag can be read.

Figure 17-3 CMP1 and CMP2 structure diagram




### 17.2.4 CMP3

CMP3 supports optional hysteresis, optional internal N-terminal bias, and digital filtering at the output. Its P-side channel can be input by GPIO or connected with OPA in the chip. Its N-terminal channel is input by GPIO; The voltage comparison result OUT0 is analog output by GPIO, and OUT1 and OUT2 are push-pull output through GPIO port. In addition, inside the chip, the output channel of CMP3 is connected to the four channels of TIM2 for capturing the trigger, and also connected to the BKIN channel of TIM1 as the brake source of TIM1.

In the CMP3_CFGR register: configure the EN bit to enable the CMP3 module; Configure the PSEL/NSEL bit, and select the positive/negative input channel of CMP3; Configure HYS bit and select the size of hysteresis voltage; Configure DACDAT[3:0] bits, and select DAC output voltage of CMP3 module; Configure OEN bit to enable CMP3 output to IO; Configure the MODE_IOSEL bit, and select the output of CMP3 to connect to IO; Or configure CMP3_BK_EN bit of CMP_CTLR register to connect the output to the brake input of TIM1.

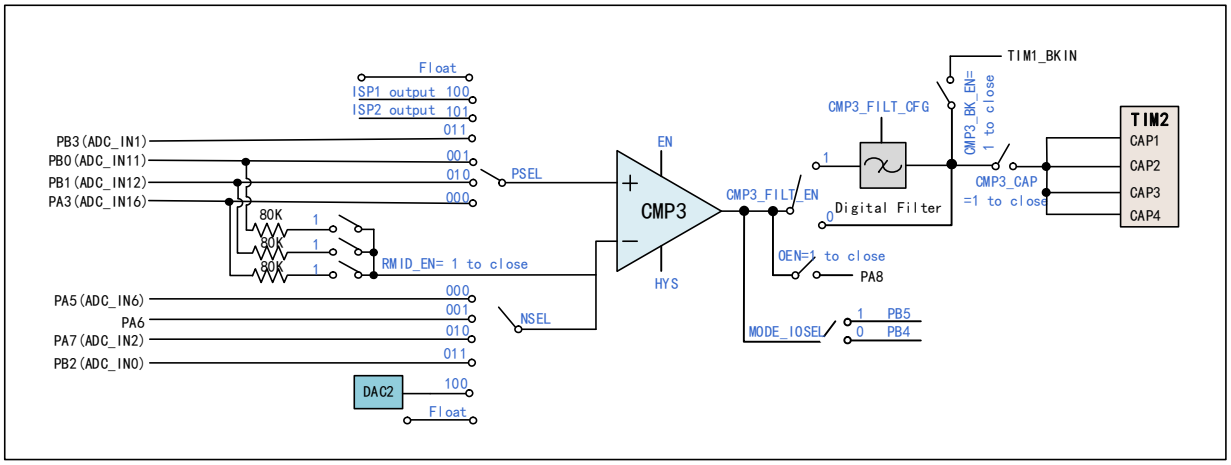
In the CMP_CTLR register: configure the CMP3_FILT_EN bit to enable the CMP3 output digital filtering function; configure the CMP3_FILT_CFG bit to select the CMP3 output digital sampling interval, with the sampling interval set to be greater than the noise width and less than twice the noise width; configure the CMP3_CAP bit to enable the CMP3 comparison output to be internally connected to the Timer2's 4-channel function; configure the CMP3_TRG_SRC bit to select the CMP3 Timer2 channel switching hardware trigger source, and set the

CMP3_TRG_GATE bit to turn on/off the CMP3 Timer2 channel hardware trigger channel switching; or configure the CMP3_COM_MODE bit to enable the CMP3 Continuous Compare mode; configure the CMP3_INT_EN bit to enable the CMP3 Continuous Compare mode; configure the CMP3_CAP bit to enable the CMP3 Compare output to be internally connected to the CMP3 Compare output to be internally connected to the CMP3 Compare output. INT_EN bit, can generate CMP3 output interrupt, and can be read through the CMP_STATR register CMP3_CHOUTx bit, CMP3 each channel for switching the comparison output results, that is, the CMP3 output is high level interrupt flag.

In the CMP3_SW register: configuring the CH_NUM bit configures the number of channels to be captured by Timer 2's channel and the output polarity of the corresponding channel via the OUT_POL_CHx bit.

In the CMP_STATR register: the CMP3_OUT/CMP3_OUT_FILT bits allow you to read the output result of the CMP3 filtering function being turned off/on, and you can read the comparison output result of the switching carried out by each channel of the CMP3, i.e., the interrupt flag of the CMP3 output going high, through the CMP3_CHOUTx bit.

Figure 17-4 CMP3 structure diagram




## 17.2.5 AC Small Signal Amplification Decoder (QII1/QII2)

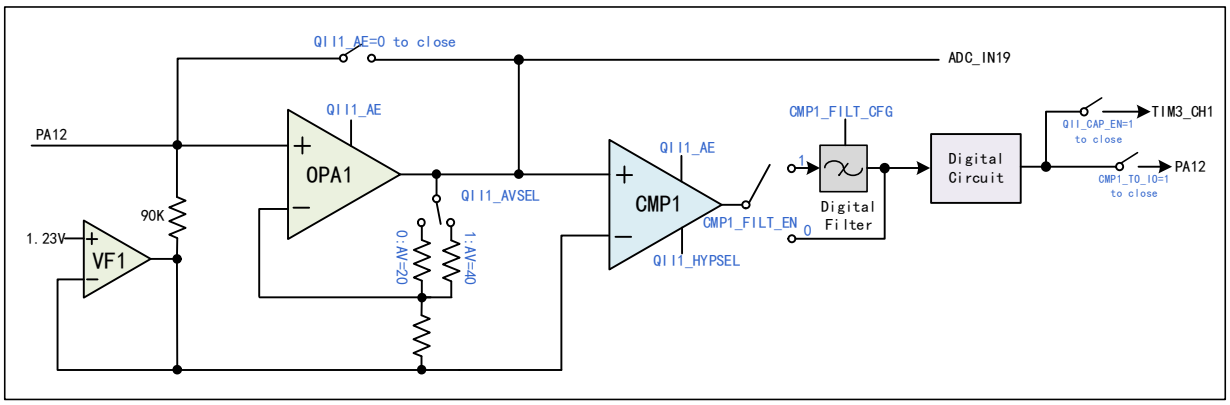
The AC small signal amplifier-decoder (QII) consists of a pre-stage adjustable gain amplifier OPA and a post-stage voltage comparator CMP. the input AC small signal is amplified by the OPA and shaped into a digital signal using the CMP, which is digitally filtered and then decoded, or the amplified signal can be fed into the ADC for decoding.

In the QII_CFGR register: configure the QII1_AE bit to enable the QII1 module; configure the QII1_HYPSEL bit to select the comparator hysteresis voltage of the QII1 module; and configure the QII1_AVSEL bit to select the gain size of the QII1 module op amp.

In the CMP_CTLR register: configure the CMP1_FILT_EN bit to enable the CMP1 output digital filtering function; configure the CMP1_FILT_CFG[4:0] bits to select the CMP1 output digital sampling interval, with the sampling interval set to be greater than the noise width and less than twice the noise width; configure the CMP1_TO_IO bit to enable the CMP1 output to the IO; configure the QII_CAP_EN bit to enable the QII outputs to be sent internally directly to TIM3 channel 1 for capture.

In the CMP_STATR register: with the CMP1_OUT/CMP1_OUT_FILT bits, you can read the output result of the CMP1 filter function off/on.

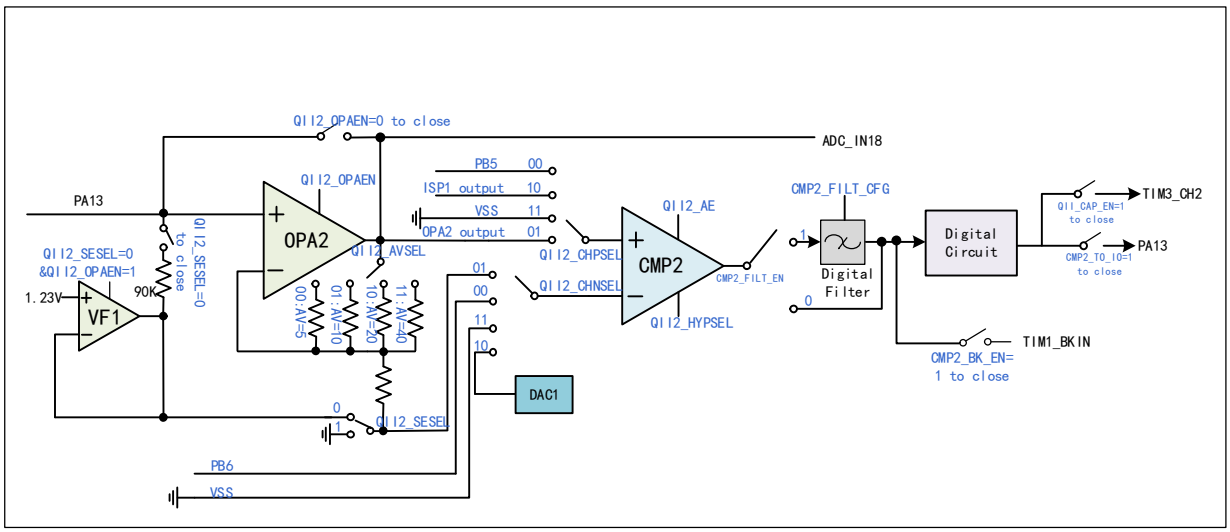
Figure 17-5 QII1 structure diagram




In the CMP_CTLR register: configure the CMP2_FILT_EN bit to enable the CMP2 output digital filtering function; configure the CMP2_FILT_CFG[4:0] bits to select the CMP2 output digital sampling interval, and the sampling interval setting should be greater than the noise width and less than twice the noise width; configure the CMP2_TO_IO bit to enable the CMP2 output Configure CMP2_TO_IO bit to enable CMP2 output to connect to IO function internally; Configure QII_CAP_EN bit to enable QII output to send to TIM3 channel 1 capture directly internally; Configure CMP2_BK_EN bit to enable CMP2 output to connect to Timer1 brake input; Configurable CMP2_INT_EN bit to enable CMP2 output interrupt.

In CMP_STATR register: the output result of CMP2 filtering function closing/opening can be read through CMP2_OUT/CMP2_OUT_FILT bits; With the CMP2_COMIF bit, the interrupt flag that CMP2 output is high level can be read.

Figure 17-6 QII2 structure diagram




## 17.2.6 Differential Input Current Sampling (ISP1/ISP2)

Differential input current sampling (ISP) amplifies the weak voltage signal collected by external resistor by closed-loop amplifier OPA, and sends the result to ADC or comparator through ADC channel. The ADC sampling result can be used for upper limit comparison, calculation and analysis, etc.

ISP1 and ISP2 support differential or single-ended applications. When differential applications are used, ISP1 and ISN1/PA8 pins are one pair of differential inputs, and ISP2/PA10 and ISN2/PA11 pins are the other pair of differential inputs; while when single-ended applications are used, the negative input ISN is not required, and the

ISN1/PA8 and ISN2/PA11 pins can be used for any purpose at that time, such as ADC or GPIO.

In the ISP_CTLR register: configure the ISPx_EN bit to enable the ISPx module; configure the ISPx_GAIN[1:0] bits to enable the magnitude of the ISPx module op-amp gain; configure the ISPx_VFBIAS bit to select the ISPx module reference voltage; configure the ISPx_NSEL bit to configure the module's negative end input; configure the ISPx_SEL IO bit, ISPx module outputs to ADC; configure QDETx_EN bit to enable ISPx module Q value detection; configure DETx_PD30K bit to enable ISPx module Q value detection of 30K resistor; configure QDETx_VBSEL bit to enable ISPx module Q value detection of reference voltage.

Figure 17-7 ISP1 structure diagram

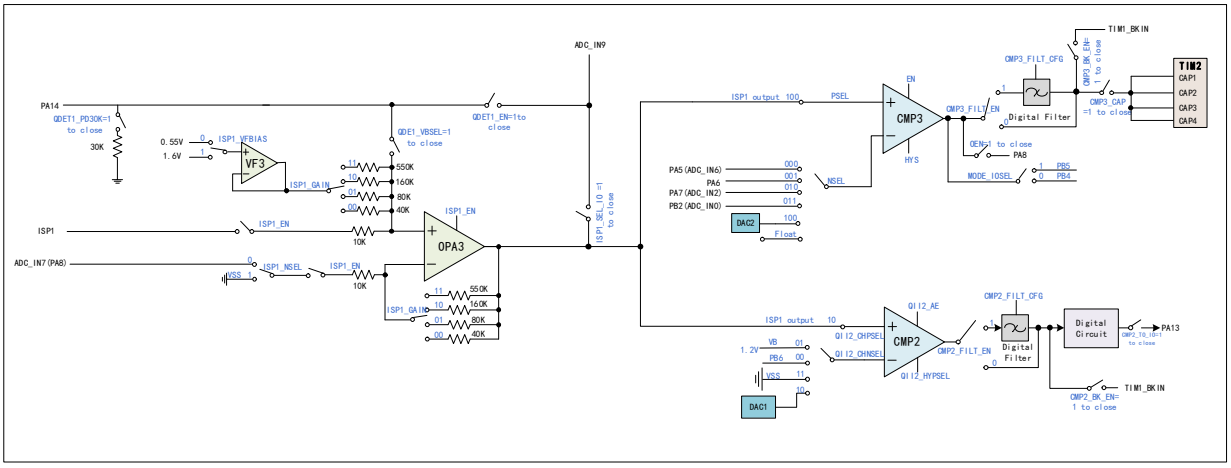
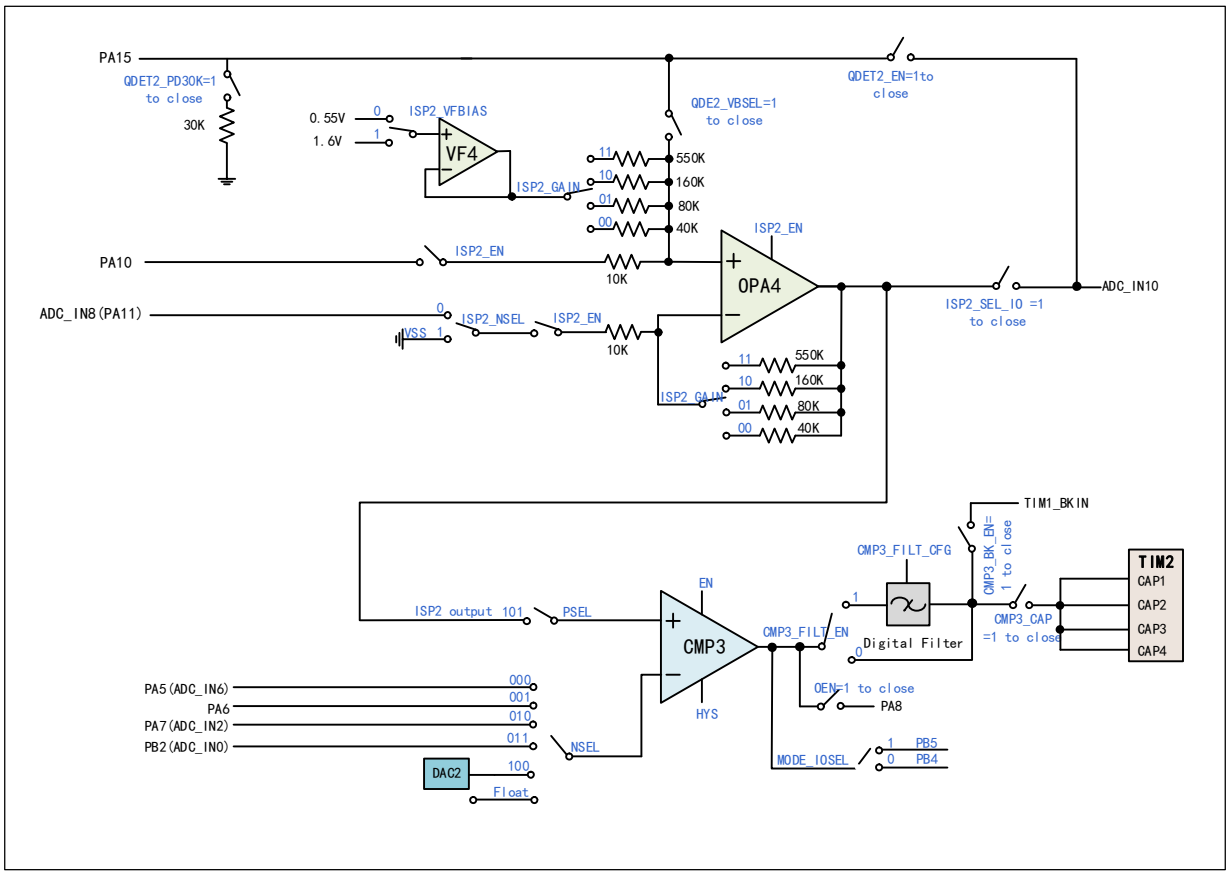



Figure 17-8 ISP2 structure diagram



## 17.3 Register Description

Table 17-4 OPA-related registers list

| Name | Offset address | Description | Reset value |
|---|---|---|---|
| R32_QII_CFGR | 0x40024800 | QII configuration register | 0x00000000 |
| R32_ISP_CTLR | 0x40024804 | ISP control register | 0x00000000 |
| R32_CMP_CTLR | 0x40024808 | CMP control register | 0x00000000 |
| R32_CMP3_CFGR | 0x4002480C | CMP3 configuration register | 0x0000007E |
| R32_CMP3_SW | 0x40024810 | CMP3 switch register | 0x00000000 |
| R32_CMP_STATR | 0x40024814 | CMP status register | 0x000XXXXX |

### 17.3.1 QII Configuration Register (QII_CFGR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | QII2_CHPSEL[1:0] | | QII2_CHNSEL[1:0] | | Reserved | QII2_DAC[3] |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QII2_DAC[2:0] | | | QII2_DACEN | Reserved | QII2_HYPSEL[2:0] | | | QII2_AVSEL[1:0] | | QII2_SESEL | QII2_OPAEN | QII2_AE | QII1_AVSEL | QII1_HYPSEL | QII1_AE |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:22] | Reserved | RO | Reserved | 0 |
| [21:20] | QII2_CHPSEL[1:0] | RW | QII2 module CMP positive input selection:<br>00: PB5;<br>01: Internal QII2 OPA output voltage;<br>10: ISP1 output;<br>11: VSS. | 0 |
| [19:18] | QII2_CHNSEL[1:0] | RW | QII2 module CMP negative input selection:<br>00: PB6;<br>01: Internal QII2 decode bias points;<br>10: Internal 4-bit DAC output;<br>11: VSS. | 0 |
| 17 | Reserved | RO | Reserved | 0 |
| [16:13] | QII2_DAC[3:0] | RW | QII2 module DAC output voltage selection (0.2V/bit):<br>0000: 0.1V;<br>0001: 0.3V;<br>0010: 0.5V;<br>……<br>1111: 3.1V.<br>*Note: $V_{DD33}$=3.3V.* | 0 |
| 12 | QII2_DACEN | RW | QII2 module DAC enable:<br>1: Enable;<br>0: Disable. | 0 |
| 11 | Reserved | RO | Reserved | 0 |
| [10:8] | QII2_HYPSEL[2:0] | RW | QII2 module comparator hysteresis selection:<br>000: 0mV;<br>001: 10mV;<br>010: 20mV;<br>011: 40mV; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 100: 60mV;<br>101: 80mV;<br>110: 100mV;<br>111: 200mV. | |
| [7:6] | QII2_AVSEL[1:0] | RW | QII2 module OPA gain selection:<br>00: 5 times;<br>01: 10 times;<br>10: 20 times;<br>11: 40 times. | 0 |
| 5 | QII2_SESEL | RW | QII2 module single-ended enlargement mode selection<br>1: Single-ended enlargement mode;<br>0: QII mode. | 0 |
| 4 | QII2_OPAEN | RW | QII2 module internal OPA enable:<br>1: Enable;<br>0: Disable. | 0 |
| 3 | QII2_AE | RW | QII2 module CMP enable:<br>1: QII2 module CMP valid;<br>0: PA13 as common IO. | 0 |
| 2 | QII1_AVSEL | RW | QII1 module OPA gain selection:<br>1: 40 times;<br>0: 20 times. | 0 |
| 1 | QII1_HYPSEL | RW | QII1 module CMP hysteresis selection:<br>1: 200mV;<br>0: 100mV. | 0 |
| 0 | QII1_AE | RW | QII1 module CMP enable:<br>1: QII1 module CMP valid;<br>0: PA12 as common IO. | 0 |

### 17.3.2 ISP Control Register (ISP_CTLR)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | QDET2_VBSEL | QDET2_PD30K | QDET2_EN | ISP2_SEL_IO | ISP2_NSEL | ISP2_VFBIAS | ISP2_GAIN[1:0] | | ISP2_EN |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | QDET1_VBSEL | QDET1_PD30K | QDET1_EN | ISP1_SEL_IO | ISP1_NSEL | ISP1_VFBIAS | ISP1_GAIN[1:0] | | ISP1_EN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:25] | Reserved | RO | Reserved | 0 |
| 24 | QDET2_VBSEL | RW | ISP2 module Q value detection reference voltage enable:<br>1: On;<br>0: Off. | 0 |
| 23 | QDET2_PD30K | RW | ISP2 module Q value detection 30K resistance enable:<br>1: On;<br>0: Off. | 0 |
| 22 | QDET2_EN | RW | ISP2 module Q value detection enable:<br>1: On;<br>0: Off. | 0 |

| | | | | |
|---|---|---|---|---|
| 21 | ISP2_SEL_IO | RW | ISP2 module output to ADC switch:<br>1: On;<br>0: Off. | 0 |
| 20 | ISP2_NSEL | RW | ISP2 module negative terminal selection:<br>1: VSS;<br>0: ISN2 (PA11). | 0 |
| 19 | ISP2_VFBIAS | RW | ISP2 module reference voltage selection:<br>1: 1.6V;<br>0: 0.55V. | 0 |
| [18:17] | ISP2_GAIN[1:0] | RW | ISP2 module OPA gain selection:<br>00: 4 times;<br>01: 8 times;<br>10: 16 times;<br>11: 55 times. | 0 |
| 16 | ISP2_EN | RW | ISP2 module enable:<br>1: Enable;<br>0: Disable. | 0 |
| [15:9] | Reserved | RO | Reserved | 0 |
| 8 | QDET1_VBSEL | RW | ISP1 module Q value detection reference voltage enable:<br>1: On;<br>0: Off. | 0 |
| 7 | QDET1_PD30K | RW | ISP1 module Q value detection 30K resistor enable:<br>1: On;<br>0: Off. | 0 |
| 6 | QDET1_EN | RW | ISP1 module Q value detection enable:<br>1: On;<br>0: Off. | 0 |
| 5 | ISP1_SEL_IO | RW | ISP1 module output to ADC switch:<br>1: On;<br>0: Off. | 0 |
| 4 | ISP1_NSEL | RW | ISP1 module negative terminal selection:<br>1: VSS;<br>0: ISN1 (PA8). | 0 |
| 3 | ISP1_VFBIAS | RW | ISP1 module reference voltage selection:<br>1: 1.6V;<br>0: 0.55V. | 0 |
| [2:1] | ISP1_GAIN[1:0] | RW | ISP1 module OPA gain selection:<br>00: 4 times;<br>01: 8 times;<br>10: 16 times;<br>11: 55 times. | 0 |
| 0 | ISP1_EN | RW | ISP1 module enable:<br>1: Enable;<br>0: Disable. | 0 |

### 17.3.3 CMP Control Register (CMP_CTLR)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | CMP3_CAP | CMP1_TO_IO | CMP2_TO_IO | CMP3_INT_EN | CMP3_BK_EN | CMP3_FILT_CFG[4:0] | | | | | CMP3_FILT_EN | CMP3_COM_MODE | QII_CAP_EN | CMP3_TRG_GATE | CMP3_TRG_SRC[2] |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMP3_TRG_SRC[1:0] | | CMP2_INT_EN | CMP2_BK_EN | CMP2_FILT_CFG[4:0] | | | | | CMP2_FILT_EN | CMP1_FILT_CFG[4:0] | | | | | CMP1_FILT_EN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31 | Reserved | RO | Reserved | 0 |
| 30 | CMP3_CAP | RW | CMP3 compare outputs are internally connected to the 4 channels of Timer2 Enable:<br>1: The compare outputs of the 4 channels of CMP3 are connected to the 4 channels of the timer correspondingly (Example: Switching channel 1 compare output is connected to the timer channel 1) for capture triggering;<br>0: The function is turned off. | 0 |
| 29 | CMP1_TO_IO | RW | CMP1 push-pull output internally connected to IO function:<br>1: CMP1 output connected to PA12;<br>0: Function off. | 0 |
| 28 | CMP2_TO_IO | RW | CMP2 push-pull output internally connected to IO function:<br>1: CMP2 output connected to PA13;<br>0: Function off. | 0 |
| 27 | CMP3_INT_EN | RW | CMP3 output interrupt enable bit:<br>1: Turn on the interrupt enable, an interrupt will be generated when the output of CMP3 changes from low to high;<br>0: Turn off the interrupt enable. | 0 |
| 26 | CMP3_BK_EN | RW | CMP3 output connected to Timer1 brake input enable:<br>1: CMP3 output connected to advanced timer brake input;<br>0: Function off.<br>*Note: The brake signal generated by CMP3 has a higher priority than CMP2.* | 0 |
| [25:21] | CMP3_FILT_CFG [4:0] | RW | CMP3 Output digital sample interval configuration (375ns/bit):<br>00000: 375ns;<br>.....<br>11111: 12us.<br>Note: The sample interval is set to be greater than the width of the noise and less than twice the width of the noise. | 0 |
| 20 | CMP3_FILT_EN | RW | CMP3 output digital filter enable:<br>1: Enable;<br>0: Disable. | 0 |
| 19 | CMP3_COM_MODE | RW | CMP3 Continuous Compare Mode Enable:<br>1: Enable CMP3 to compare in real time;<br>0: Disable CMP3 to compare in real time. | 0 |
| 18 | QII_CAP_EN | RW | QII outputs are internally sent directly to the TIM3 capture enable bit:<br>1: QII outputs are internally sent directly to the TIM3 capture enable, where QII1 is output to TIM3_CH1 and QII2 is output to TIM3_CH2.<br>0: Off. | 0 |
| 17 | CMP3_TRG_GATE | RW | Hardware-triggered channel switching control for CMP3 internally connected timer 2 channels: | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Disable the hardware-triggered channel switching function of CMP3 timer channels;<br>0: Open the hardware-triggered channel switching function of CMP3 timer channels.<br>*Note: Before modifying CMP3_TRG_SRC, it is recommended to set this position 1 and clear this bit after CMP3_TRG_SRC modification is finished.* | |
| [16:14] | CMP3_TRG_SRC[2:0] | RW | The CMP3 is internally connected to the timer 2-channel hardware toggle trigger source selection:<br>000: CC4 event of timer 1;<br>001: CC5 event of timer 1;<br>010: CC1 event of timer 2;<br>011: CC2 event of timer 2;<br>100: CC3 event of timer 2;<br>101: CC4 event of timer 2;<br>110: CC1 event of timer 3;<br>111: CC2 event of timer 3. | 0 |
| 13 | CMP2_INT_EN | RW | CMP2 output interrupt enable bit:<br>1: Turn on interrupt enable;<br>0: Turn off interrupt enable. | 0 |
| 12 | CMP2_BK_EN | RW | CMP2 output connected to Timer1 brake input enable:<br>1: CMP2 output connected to advanced timer brake input;<br>0: Function off.<br>*Note: The brake signal generated by CMP3 has a higher priority than CMP2.* | 0 |
| [11:7] | CMP2_FILT_CFG[4:0] | RW | CMP2 Output digital sample interval configuration (375ns/bit):<br>00000: 375ns;<br>.....<br>11111: 12us.<br>Note: The sample interval is set to be greater than the width of the noise and less than twice the width of the noise. | 0 |
| 6 | CMP2_FILT_EN | RW | CMP2 output digital filter enable:<br>1: Enable;<br>0: Disable. | 0 |
| [5:1] | CMP1_FILT_CFG[4:0] | RW | CMP1 Output digital sample interval configuration (375ns/bit):<br>00000: 375ns;<br>.....<br>11111: 12us.<br>*Note: The sample interval is set to be greater than the width of the noise and less than twice the width of the noise.* | 0 |
| 0 | CMP1_FILT_EN | RW | CMP1 output digital filter enable:<br>1: Enable;<br>0: Disable. | 0 |

### 17.3.4 CMP3 Configuration Register (CMP3_CFGR)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | MODE_IOSEL |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OE | OEN | DACDAT[3:0] | | | | RMID_EN | HYS[1:0] | | NSEL[2:0] | | | PSEL[2:0] | | | EN |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:17] | Reserved | RO | Reserved | 0 |
| 16 | MODE_IOSEL | RW | CMP3 push-pull output pin selection when connected to IO:<br>1: PB5;<br>0: PB4. | 0 |
| 15 | OE | RW | IO output enable when CMP3 push-pull output is connected to IO:<br>1: Enable;<br>0: Disable. | 0 |
| 14 | OEN | RW | CMP3 analog output to IO enable:<br>1: Enable;<br>0: Disable. | 0 |
| [13:10] | DACDAT[3:0] | RW | CMP3 module DAC output selection bit (0.2V/bit):<br>0000: 0.1V;<br>……<br>1111: 3.1V.<br>*Note: $V_{DD33}$=3.3V。* | 0 |
| 9 | RMID_EN | RW | CMP3 virtual center point enable bit:<br>1: On;<br>0: Off. | 0 |
| [8:7] | HYS[1:0] | RW | CMP3 hysteresis voltage selection bit:<br>00: 0Mv;<br>01: 10mV;<br>10: 20mV;<br>11: 40mV. | 0 |
| [6:4] | NSEL[2:0] | RW | CMP3 negative channel selection bit:<br>000: PA5;<br>001: PA6;<br>010: PA7;<br>011: PB2;<br>100: DAC output;<br>Others: Reserved. | 0x7 |
| [3:1] | PSEL[2:0] | RW | CMP3 positive channel selection bit:<br>000：PA3；<br>001：PB0；<br>010：PB1；<br>011：PB3；<br>100：ISP1 output;<br>101：ISP2 output;<br>Others: Reserved. | 0x7 |
| 0 | EN | RW | CMP3 enable:<br>1: On;<br>0: Off. | 0 |

### 17.3.5 CMP3 Switch Register (CMP3_SW)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| Reserved | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | OUT_POL_CH4 | OUT_POL_CH3 | OUT_POL_CH2 | OUT_POL_CH1 | CH_NUM[1:0] | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:6] | Reserved | RO | Reserved | 0 |
| 5 | OUT_POL_CH4 | RW | Polarity configuration bits for CMP3 internal outputs to Timer 2 capture channel 4:<br>1: Timer capture signal flipped;<br>0: Timer capture signal unchanged.<br>*Note: The comparator output signal does not change.* | 0 |
| 4 | OUT_POL_CH3 | RW | Polarity configuration bits for CMP3 internal outputs to Timer 2 capture channel 3:<br>1: Timer capture signal flipped;<br>0: Timer capture signal unchanged.<br>*Note: The comparator output signal does not change.* | 0 |
| 3 | OUT_POL_CH2 | RW | Polarity configuration bits for CMP3 internal outputs to Timer 2 capture channel 2:<br>1: Timer capture signal flipped;<br>0: Timer capture signal unchanged.<br>*Note: The comparator output signal does not change.* | 0 |
| 2 | OUT_POL_CH1 | RW | Polarity configuration bits for CMP3 internal outputs to Timer 2 capture channel 1:<br>1: Timer capture signal flipped;<br>0: Timer capture signal unchanged.<br>*Note: The comparator output signal does not change.* | 0 |
| [1:0] | CH_NUM[1:0] | RW | Number of channels captured by the CMP3 internal outputs to Timer 2:<br>00: Captured using Timer 2 channel 1 only;<br>01: Captured cyclically using Timer 2 channel 1 + channel 2;<br>10: Captured cyclically using Timer 2 channel 1 + channel 2 + channel 3;<br>11: Captured cyclically using Timer 2 channel 1 + channel 2 + channel 3 + channel 4. | 0 |

### 17.3.6 CMP Status Register (CMP_STATR)

Offset address: 0x14

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | CMP3_CHOUT3 | CMP3_CHOUT2 | CMP3_CHOUT1 | CMP3_CHOUT0 | CMP3_OUT_FILT | CMP3_OUT |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | CMP2_COMIF | CMP2_OUT_FILT | CMP2_OUT | Reserved | | | | | | CMP1_OUT_FILT | CMP1_OUT |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:22] | Reserved | RO | Reserved | 0 |
| 21 | CMP3_CHOUT3 | RW1 | Comparison output result of the 4th channel of CMP3 internal connection timer 2:<br>1: CMP3 output is high;<br>0: CMP3 output is low.<br>Write 1 to clear zero, write 0 to invalid. | 0 |
| 20 | CMP3_CHOUT2 | RW1 | Comparison output result of the 3rd channel of CMP3 internal connection timer 2:<br>1: CMP3 output is high;<br>0: CMP3 output is low.<br>Write 1 to clear zero, write 0 to invalid. | 0 |
| 19 | CMP3_CHOUT1 | RW1 | Comparison output result of the 2nd channel of CMP3 internal connection timer 2:<br>1: CMP3 output is high;<br>0: CMP3 output is low.<br>Write 1 to clear zero, write 0 to invalid. | 0 |
| 18 | CMP3_CHOUT0 | RW1 | Comparison output result of the 1st channel of CMP3 internal connection timer 2:<br>1: CMP3 output is high;<br>0: CMP3 output is low.<br>Write 1 to clear zero, write 0 to invalid. | 0 |
| 17 | CMP3_OUT_FILT | RO | CMP3 turns on the output result of the filtering function. | x |
| 16 | CMP3_OUT | RO | CMP3 turns off the output result of the filtering function. | x |
| [15:11] | Reserved | RO | Reserved | 0 |
| 10 | CMP2_COMIF | RW1 | CMP2 output is high level interrupt flag bit:<br>1: Indicates that the comparison result of CMP2 turns from low to high.<br>Write 1 to clear, write 0 is invalid. | 0 |
| 9 | CMP2_OUT_FILT | RO | CMP2 turns on the output result of the filtering function. | x |
| 8 | CMP2_OUT | RO | CMP2 turns off the output result of the filtering function. | x |
| [7:2] | Reserved | RO | Reserved。 | 0 |
| 1 | CMP1_OUT_FILT | RO | CMP1 turns on the output result of the filtering function. | x |
| 0 | CMP1_OUT | RO | CMP1 turns off the output result of the filtering function. | x |

# Chapter 18 Electronic Signature (ESIG)

The electronic signature contains chip identification information: The capacity of the flash memory area and the unique identification. It is programmed into the system storage area of the memory module by the manufacturer at the factory, and can be read by SWD (SDI) or application code.

## 18.1 Functional Description

Flash memory area capacity: Indicates the available size of the current chip user application program.

Unique ID: 96-bit binary code, unique to any microcontroller; users can only read and access but cannot modify it. This unique identification information can be used as the security password, encryption key, product serial number, etc. of the microcontroller (Product) to improve the system security mechanism or indicate identity information.

Users of the above content can conduct read access according to 8/16/32 bits.

## 18.2 Register Description

Table 18-1 ESIG registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_ESIG_FLACAP | 0x1FFFF3A0 | Flash capacity register | 0xXXXX |
| R32_ESIG_UNIID1 | 0x1FFFF3A8 | UID register 1 | 0xXXXXXXXX |
| R32_ESIG_UNIID2 | 0x1FFFF3AC | UID register 2 | 0xXXXXXXXX |
| R32_ESIG_UNIID3 | 0x1FFFF3B0 | UID register 3 | 0xXXXXXXXX |

### 18.2.1 Flash Capacity Register (ESIG_FLACAP)

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | F_SIZE[15:0] | RO | Flash memory capacity in Kbyte.<br>Example: 0x0080 = 128K bytes | X |

### 18.2.2 UID Register (ESIG_UNIID1)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U_ID[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| U_ID[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | U_ID[31:0] | RO | UID's bits 0-31. | X |

### 18.2.3 UID Register (ESIG_UNIID2)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U_ID[63:48] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U_ID[47:32] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | U_ID[63:32] | RO | UID's bits 32-63. | X |

### 18.2.4 UID Register (ESIG_UNIID3)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U_ID[95:80] | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U_ID[79:64] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | U_ID[95:64] | RO | UID's bits 64-95. | X |

# Chapter 19 Flash Memory and User Option Bytes

## 19.1 Flash Memory Organization

The internal flash memory organization structure of the chip is as follows:

Table 19-1 Flash memory organization structure

| Block | Name | Address Range | Size(byte) |
|---|---|---|---|
| Main memory | Page 0 | 0x08000000 – 0x0800007F | 128 |
| | Page 1 | 0x08000080 – 0x080000FF | 128 |
| | Page 2 | 0x08000100 – 0x0800017F | 128 |
| | Page 3 | 0x08000180 – 0x080001FF | 128 |
| | Page 4 | 0x08000200 – 0x0800027F | 128 |
| | Page 5 | 0x08000280 – 0x080002FF | 128 |
| | Page 6 | 0x08000300 – 0x0800037F | 128 |
| | Page 7 | 0x08000380 – 0x080003FF | 128 |
| | … | … | … |
| | Page 511 | 0x0800FF80 – 0x0800FFFF | 128 |
| Information block | User-defined information store | 0x1FFFF000 – 0x1FFFF1FF | 512 |
| | User option bytes | 0x1FFFF300 – 0x1FFFF37F | 128 |
| | Manufacturer configuration word | 0x1FFFF380 – 0x1FFFF3FF | 128 |

*Notes:*

*(1) The main memory area is used for storing user's application programs, and is write-protected in units of 4K bytes (32 pages); Except for the factory lock of the "Manufacturer Configuration Word" area, it is inaccessible to users, and other areas can be operated by users under certain conditions.*

*(2) The user's nonvolatile data storage area can only be operated by the WCH-LinkUtility software tool.*

## 19.2 Flash Memory Programming and Safety

1) Programming/erasing mode

- Fast programming: this method uses the page operation method (Recommended). After a specific sequence of unlocking, a single 256 byte programming and 256 byte erasure, 1K byte erasure and whole chip erasure are performed.

2) Security - Protection against Illegal Access (Read, write, erase)

- Page write protection
- Read protection

Under the read protection state:

1） The main memory pages 0-31 (4K bytes) are automatically write-protected and are not controlled by the FLASH_WPR register; when the read protection status is released, all main memory pages will be controlled by the FLASH_WPR register.

2） The main memory cannot be erased or programmed in the system boot code area, SWD or SDI mode, and RAM area, except for the entire chip erasure. User-selected word area can be erased or programmed. If you try to release the read protection (Program user word), the chip will automatically erase the entire user area.

*Note: When programming/erasing operations of flash memory are made, the internal RC oscillator (HSI) must be switched on.*

## 19.3 Register Description

Table 19-2 FLASH registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_FLASH_ACTLR | 0x40022000 | Control register | 0x00000010 |
| R32_FLASH_KEYR | 0x40022004 | FPEC key register | 0xXXXXXXXX |
| R32_FLASH_OBKEYR | 0x40022008 | OBKEY register | 0xXXXXXXXX |
| R32_FLASH_STATR | 0x4002200C | Status register | 0x00008000 |
| R32_FLASH_CTLR | 0x40022010 | Configuration register | 0x00008080 |
| R32_FLASH_ADDR | 0x40022014 | Address register | 0xXXXXXXXX |
| R32_FLASH_OBR | 0x4002201C | Option byte register | 0x00000XXX |
| R32_FLASH_WPR | 0x40022020 | Write protection register | 0x0000XXXX |
| R32_FLASH_MODEKEYR | 0x40022024 | Extension key register | 0xXXXXXXXX |
| R8_FLASH_CTLR2 | 0x4002202F | Configuration register 2 | 0xXX |

### 19.3.1 Access Control Register (FLASH_ACTLR)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | LATENCY [1:0] | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:2] | Reserved | RO | Reserved. | 0x00000002 |
| [1:0] | LATENCY[1:0] | RW | Time delay. Ratio of system clock (SYSCLK) to flash memory access time:<br>When the maximum operating temperature of the chip is 105°C:<br>000: 0 wait (0≤SYSCLK≤24MHz);<br>001: 1 wait (24MHz<SYSCLK≤48MHz);<br>010: 2 wait (48MHz<SYSCLK≤72MHz);<br>011: 3 wait (72MHz<SYSCLK≤96MHz);<br>Other: Reserved.<br>When the maximum operating temperature of the chip is 125°C:<br>000: 0 wait (0≤SYSCLK≤20MHz);<br>001: 1 wait (20MHz<SYSCLK≤40MHz);<br>010: 2 wait (40MHz<SYSCLK≤60MHz);<br>011: 3 wait (60MHz<SYSCLK≤80MHz);<br>Other: Reserved. | 0 |

### 19.3.2 FPEC Key Register (FLASH_KEYR)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| FKEYR[31:16] | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FKEYR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | FKEYR[31:0] | WO | FPEC keys, unlock keys for entering FPEC include:<br>RDPRT = 0x000000A5;<br>KEY1 = 0x45670123;<br>KEY2 = 0xCDEF89AB. | X |

### 19.3.3 OBKEY Register (FLASH_OBKEYR)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OBKEYR[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OBKEYR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | OBKEYR[31:0] | WO | Option byte key for entering the select word key to unlock OBWRE.<br>KEY1 = 0x45670123;<br>KEY2 = 0xCDEF89AB.<br>*(Note: Need to unlock FLASH first)* | X |

### 19.3.4 Status Register (FLASH_STATR)

Offset address: 0x0C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | ECCERRIF | NOERR | ERRCOR | ERR_FATAL | Reserved |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | EOP | WRPRTERR | Reserved | | | BSY |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:21] | Reserved | RO | Reserved | 0 |
| 20 | ECCERRIF | RW1 | FLASH module ECC check error interrupt flag.<br>1: Error in ECC check;<br>0: Error-free ECC check.<br>Write 1 to clear, write 0 is invalid. | 0 |
| 19 | NOERR | RO | Error-free ECC check of FLASH module:<br>1: Error-free ECC check;<br>0: No effect. | 0 |
| 18 | ERRCOR | RO | Error in ECC check of FLASH module, error correction succeeded. | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Error in ECC check, error correction succeeded;<br>0: No effect. | |
| 17 | ERR_FATAL | RO | Error in ECC check of FLASH module, error correction failed.<br>1: Error in ECC check, error correction failed.<br>0: No effect. | 0 |
| [16:6] | Reserved | RO | Reserved | 0 |
| 5 | EOP | RW1 | Indicates the end of the operation, write 1 to clear.<br>The hardware is set every time it is successfully erased or programmed. | 0 |
| 4 | WRPRTERR | RW1 | Indicates write protection error, write 1 to clear.<br>If you program a write-protected address, the hardware will set it. | 0 |
| [3:1] | Reserved | RO | Reserved. Must remain in the cleared state' 0'. | 0 |
| 0 | BSY | RO | Indicates busy status:<br>1: Indicates that the flash operation is in progress;<br>0: Operation ended or an error occurred. | 0 |

*Note: When performing the programming operation, you need to make sure that the STRT bit in the FLASH_CTLR register is set to 0.*

### 19.3.5 Configuration Register (FLASH_CTLR)

Offset address: 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | ECCCORIE | ECCERRIE | BUFRST | BUFLOAD | FTER | FTPG |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FLOCK | Reserved | | EOPIE | Reserved | ERRIE | OBWRE | Reserved | LOCK | STRT | OBER | Reserved | | MER | PER | Reserved |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:22] | Reserved | RO | Reserved | 0 |
| 21 | ECCCORIE | RW | An ECC checksum error occurs but error correction is successful, and the ECC error flag is set to select.<br>1: The ECC error interrupt flag is set when ECC checksum error correction is successful;<br>0: The ECC error interrupt flag will not be set when ECC checksum error correction is successful. | 0 |
| 20 | ECCERRIE | RW | ECC error interrupt enable bit:<br>1: Enable the ECC error interrupt;<br>0: Disable the ECC error interrupt. | 0 |
| 19 | BUFRST | RW | Perform internal buffer data cleanup. | 0 |
| 18 | BUFLOAD | RW | Perform data loading into the internal buffer. | 0 |
| 17 | FTER | RW | Perform a fast page (128Byte) erase operation. | 0 |
| 16 | FTPG | RW | Perform fast programming operations. | 0 |
| 15 | FLOCK | RW1 | Fast programming lock. You can only write' 1'. When this bit is' 1', it means that the fast program/erase mode is not available. After | 1 |

| | | | | |
|---|---|---|---|---|
| | | | detecting the correct unlocking sequence, the hardware clears this bit to' 0'.<br>Set the software to 1 and re-lock. | |
| [14:13] | Reserved | RO | Reserved | 0 |
| 12 | EOPIE | RW | Operation completion interrupt control (EOP set in FLASH_STATR register):<br>1: Interrupt generation is allowed;<br>0: Interrupt generation is prohibited. | 0 |
| 11 | Reserved | RO | Reserved | 0 |
| 10 | ERRIE | RW | Error status interrupt control (PGERR/WRPRTERR set in FLASH_STATR register):<br>1: interrupt generation is allowed;<br>0: interrupt generation is prohibited. | 0 |
| 9 | OBWRE | RW0 | User option byte lock, the software clears 0:<br>1: Indicates that the user select word can be programmed for operation. Needs to be set by hardware after writing the correct sequence in the FLASH_OBKEYR register;<br>0: Re-lock the user select word after software clear 0. | 0 |
| 8 | Reserved | RO | Reserved | 0 |
| 7 | LOCK | RW1 | Lock. Only '1' can be written. When this bit is '1' it indicates that FPEC and FLASH_CTLR are locked and not writable. Hardware clears this bit to '0' after a correct unlock sequence is detected. After an unsuccessful unlock operation, this bit will not change again until the next system reset. | 1 |
| 6 | STRT | RW1 | Start. Setting 1 initiates an erase or fast programming action, and the hardware automatically clears 0 (BSY goes to '0'). | 0 |
| 5 | OBER | RW | Performs user option byte erasure | 0 |
| [4:3] | Reserved | RO | Reserved | 0 |
| 2 | MER | RW | Performs a full erase operation (erases the entire user area). | 0 |
| 1 | PER | RW | Performs a standard page (1KB) erase operation. | 0 |
| 0 | Reserved | RO | Reserved | 0 |

### 19.3.6 Address Register (FLASH_ADDR)

Offset address: 0x14

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FAR[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FAR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | FAR[31:0] | WO | The flash address, the programmed address when programming is performed, and the start address of the erase when erasing is performed. This register cannot be written when the BSY bit in the FLASH_STATR register is '1'. | X |

### 19.3.7 Option Byte Register (FLASH_OBR)

Offset address: 0x1C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | 0V_CFG | Reserved | RST_PIN_SEL | RST_MODE[1:0] | | STANDYRST | STOPRST | Reserved | RDPRT | OPTERR |

| Bit | Name | | Access | Description | Reset value |
|---|---|---|---|---|---|
| [31:10] | Reserved | | RO | Reserved | 0 |
| 9 | USER | OV_CFG | RO | Overvoltage protection detection function user configuration bits:<br>1: The overvoltage protection detection function is determined by the PWR_CTLR register SEL_IO_VHV bit;<br>0: Overvoltage protection detection is forced to be turned on by the user. | 1 |
| 8 | | Reserved | RO | Reserved | x |
| 7 | | RST_PIN_SEL | RO | Reset pin select bit:<br>1: PA15;<br>0: PC0. | 1 |
| [6:5] | | RST_MODE[1:0] | RO | Alternate is external pin reset.<br>00: Enable alternate function, power-on reset ignores pin status within 128us;<br>01: Enable alternate function, power-on reset ignores the pin status within 512us;<br>10: Enable alternate function, power-on reset ignores the pin state within 1ms;<br>11: Disable alternate function, RST is IO function.<br>*Note: Default off pin reset function.* | 11b |
| 4 | | STANDYRST | RO | Low power management reset configuration for Standby mode:<br>1: Disable low power management reset for Standby mode;<br>0: Enable the low-power management reset for Standby mode. | 1 |
| 3 | | STOPRST | RO | Low power management reset configuration for Stop mode:<br>1: Disable low power management reset for Stop mode;<br>0: Enable the low-power management reset for Stop mode. | 1 |
| 2 | | Reserved | RO | Reserved | x |
| 1 | RDPRT | | RO | Read protection status.<br>1: Indicates that the flash memory is currently read-protected;<br>0: Indicates that the current read protection of the flash memory is invalid. | x |
| 0 | OPTERR | | RO | Option byte error.<br>1: Indicates that the option byte and its | x |

| | | | | |
|---|---|---|---|---|
| | | | inverse code do not match;<br>0: Indicates that the option byte and its inverse code match. | |

*Note: USER and RDPRT are loaded from the user-selected word area after system reset.*

### 19.3.8 Write Protection Register (FLASH_WPR)

Offset address: 0x20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRP[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0 |
| [15:0] | WPR[15:0] | RO | Flash memory write-protect status.<br>1: Write protection disabled;<br>0: Write protection active.<br>Each bit represents 4K bytes (32 pages) of memory write protection status. | X |

*Note: The WPR is loaded from the user-selected word area after a system reset.*

### 19.3.9 Extension Key Register (FLASH_MODEKEYR)

Offset address: 0x24

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODEKEYR[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MODEKEYR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | MODEKEYR[31:0] | WO | Enter the following sequence to unlock the fast programming/erase mode:<br>KEY1 = 0x45670123;<br>KEY2 = 0xCDEF89AB.<br>*(Note: FLASH needs to be unlocked first)* | X |

### 19.3.10 Configuration Register 2 (FLASH_CTLR2)

Offset address: 0x2F

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RW | Reserved | x |
| [6:5] | VDD8_SEL[1:0] | RW | $V_{DD8}$ voltage selection bit:<br>00: 5V;<br>01: 8V;<br>10: 9V;<br>11: 10V. | 0 |
| [4:0] | Reserved | RW | Reserved | x |

## 19.4 Flash Operation Procedure

### 19.4.1 Read Operation

Direct addressing is in the general address space, and the user can access the content of the flash memory module and get the corresponding data through any read operation of 8/16/32-bit data.

### 19.4.2 Unlock Flash Memory

After the system reset, the flash memory controller (FPEC) and FLASH_CTLR register will be locked and cannot be accessed. The flash memory controller module can be unlocked by writing the sequence to the FLASH_KEYR register.

Unlock sequence:

1） Write KEY1 = 0x45670123 to the FLASH_KEYR register (Must operate KEY1 first);

2） Write KEY2 = 0xCDEF89AB to the FLASH_KEYR register (Must operate KEY2 secondly).

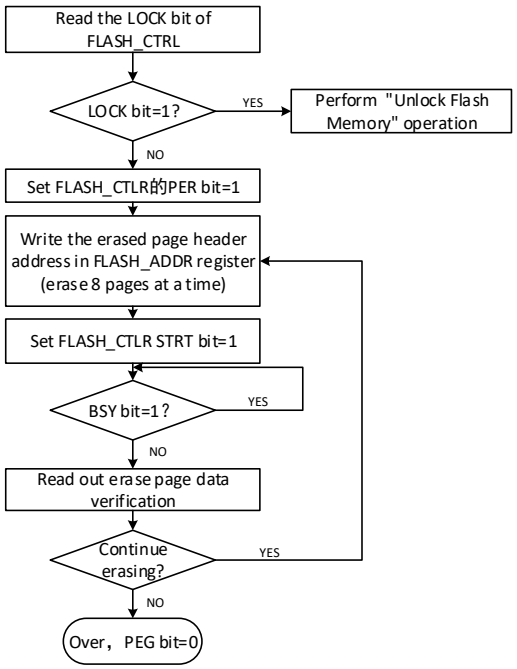
The above operations must be performed sequentially and continuously. Otherwise, it is an error operation, which will lock the FPEC module and FLASH_CTLR register and generate a bus error until the next system reset.

The flash memory controller (FPEC) and the FLASH_CTLR register can be locked again by setting the "LOCK" bit in the FLASH_CTLR register to 1.

### 19.4.3 Main Memory Standard Erase

Flash memory can be erased by standard page (1K bytes) or by whole chip.

Figure 19-1 FLASH page erase




1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to perform the "Unlock Flash" operation.

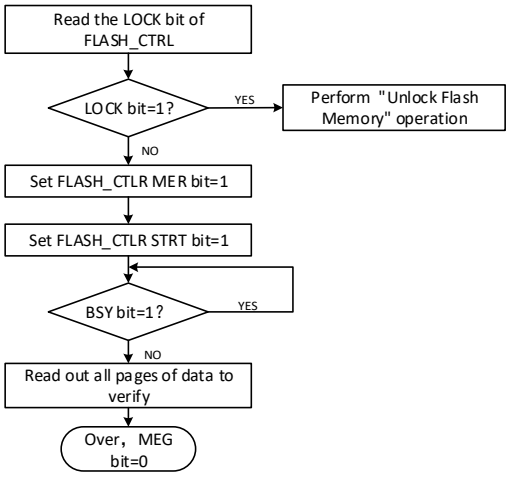
2) Set the PER bit of FLASH_CTLR register to '1' to enable the standard page erase mode.

3) Write the page header address to the FLASH_ADDR register to select the page to be erased.

4) Set the STRT bit of FLASH_CTLR register to '1' to initiate an erase action.

5) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to become '1' to indicate the end of erasure, and clear the EOP bit to 0.
6) Read the data of the erased page for verification.
7) Continue the standard page erase can repeat steps 3-5, end the erase to clear the PEG bit to 0.

Figure 19-2 FLASH whole chip erase




1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to perform the "unlock flash memory" operation.
2) Set the MER bit of FLASH_CTLR register to '1' to enable the whole chip erase mode.
3) Set the STRT bit of FLASH_CTLR register to '1' to start the erase operation.
4) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erasure, and clear the EOP bit to 0.
5) Read the data of the erased page for verification.
6) Clear the MER bit to 0.

### 19.4.4 Fast Programming Mode Unlock

Fast programming mode operation can be unlocked by writing a sequence to the FLASH_MODEKEYR register. After unlocking, the FLOCK bit in the FLASH_CTLR register will clear to 0, indicating that fast erase and programming operations can be performed. The FLASH_CTLR register can be locked again by software setting the "FLOCK" bit to 1.

Unlock sequence:
1) Write KEY1 = 0x45670123 to the FLASH_MODEKEYR register;
2) Write KEY2 = 0xCDEF89AB to the FLASH_MODEKEYR register.

The above operations must be performed sequentially and consecutively, otherwise they will be locked as errors and cannot be unlocked until the next system reset.

*Note: The fast programming operation needs to unlock the "LOCK" and "FLOCK" layers.*

### 19.4.5 Main Memory Fast Programming

Fast programming is programmed by page (128 bytes). The system has a built-in 128-byte buffer area, so it is more

efficient to save the data line to be programmed to the buffer area and perform a programming operation once.

1) Check the FLASH_CTLR register LOCK bit, if it is '1', you need to perform the "Unlock Flash Memory" operation.
2) Check the FLASH_CTLR register FLOCK bit, if it is '1', you need to perform the "Fast Programming Mode Unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to make sure there is no other programming operation in progress.
4) Set the FTPG bit of FLASH_CTLR register to '1' to enable fast page programming mode.
5) Set the BUFRST bit of the FLASH_CTLR register to perform a clear internal 128 bytes buffer operation.
6) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to become '1' to indicate the end of clearing, and clear the EOP bit to 0
7) Write 8 bytes of data to the specified address continuously (4 bytes/operation, with the address offset of 4 each time), then set the BUFLOAD bit in the FLASH_CTLR register and load it into the buffer.
8) Waiting for the BYS bit to become' 0' or the EOP bit in the FLASH_STATR register to be' 1' indicates the end of loading, and clearing the EOP bit to 0.
9) Repeat steps 7-8 for a total of 16 times, and load all 128 bytes of data into the buffer area (the addresses should be continuous after 16 rounds of operation).
10) Write the first address of the fast programming page to the FLASH_ADDR register.
11) Set the STRT bit of the FLASH_CTLR register to '1' to start fast page programming.
12) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to become '1' to indicate the completion of one fast page programming and clear the EOP bit to 0.
13) Query the FLASH_STATR register to see if there is an error or read the programmed address data checksum.
14) Continue fast page programming by repeating steps 5-13 and end programming by clearing the FTPG bit to 0.

### 19.4.6 Main Memory Fast Erase

Fast erase is performed by page (128 bytes).

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to perform the "Unlock Flash Memory" operation.
2) Check the BSY bit in the FLASH_STATR register to confirm that there are no other programming operations in progress.
3) Check the FLASH_CTLR register FLOCK bit, if it is 1, you need to perform the "fast programming mode unlock" operation.
4) Set the FTER bit in the FLASH_CTLR register to turn on the fast erase mode function.
5) Write the first address of the fast erase page to the FLASH_ADDR register.
6) Set the STAT bit in the FLASH_CTLR register to' 1' to start a quick page erase operation.
7) Waiting for the BYS bit to become' 0' or the EOP bit in the FLASH_STATR register to be' 1' indicates the end of erasing, and clearing the EOP bit to 0.
8) Query the FLASH_STATR register to see if there is any error, or read and erase the page address data verification.
9) To continue the fast page erase, you can repeat steps 5-8, and clear FTER bit at the end of the erase.

## 19.5 User Option Bytes

The user option byte is solidified in FLASH and is reloaded into the corresponding register after a system reset, and

can be erased and programmed by the user at will. The user select word information block has a total of 8 bytes (4 bytes for write protection, 1 byte for read protection, 1 byte for configuration options and 2 bytes for user data) and each bit has its own inverse code bit for verification during loading. The structure and meaning of the select word information is described below.

Table 19-3 32-bit option byte format division

| [31:24] | [23:16] | [15:8] | [7:0] |
|---|---|---|---|
| Inverse code of option bytes 1 | Option bytes 1 | Inverse code of option bytes 0 | Option bytes 0 |

Table 19-4 User option bytes information structure

| Address / Bit | [31:24] | [23:16] | [15:8] | [7:0] |
|---|---|---|---|---|
| 0x1FFFF800 | nUSER | USER | nRDPR | RDPR |
| 0x1FFFF804 | nData1 | Data1 | nData0 | Data0 |
| 0x1FFFF808 | nWRPR1 | WRPR1 | nWRPR0 | WRPR0 |

| Name/Byte | | | Description | Reset value |
|---|---|---|---|---|
| RDPR | | | Read protection control. It configures whether the code in the flash memory can be read.<br>0xA5: If this byte is 0xA5 (nRDP must be 0x5A), it means that the current code is in a non-read protected state and can be read;<br>Other values: Code read protection status, unreadable; pages 0 to 7 pages (2K) will be automatically write-protected and not controlled by WRPR0. | 0xA5 |
| USER | 7 | OV_CFG | User configuration bits for overvoltage protection detection function:<br>1: The detection function of overvoltage protection is determined by SEL_IO_VHV bit in PWR_CTLR register;<br>0: The user forcibly turns on overvoltage protection detection. | 1 |
| | 6 | Reserved | Reserved | x |
| | 5 | RST_PIN_SEL | Reset pin select bit:<br>1: PA15;<br>0: PC0. | 1 |
| | [4:3] | RST_MODE | External reset pin RST enable:<br>00: Turn on the alternate function, and ignore the pin state within 128us during power-on reset;<br>01: Turn on the alternate function, and ignore the pin state within 512us during power-on reset;<br>10: Turn on the alternate function, and ignore the pin state within 1ms during power-on reset;<br>11: The alternate function is turned off, and RST is the IO function.<br>*Note: The pin reset function is turned off by default.* | 11b |
| | 2 | STANDYRST | System reset control in Standby mode:<br>1: Disable; system is not reset when entering Standby mode;<br>0: Enable; system is reset when entering Standby mode. | 1 |
| | 1 | STOPRST | System reset control in Stop mode: | 1 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Disable; system will not be reset when entering Stop mode;<br>0: Enable; system will be reset when entering Stop mode. | |
| | 0 | Reserved | Reserved | 1 |
| Data0–Data1 | | | Saving the user's data 2 bytes. | 0xFFFF |
| WRPR0 – WRPR1 | | | Write protection control bit. Each bit is used to control the write protection status of 4K bytes in the main memory:<br>1: Disable write protection;<br>0: Enable write protection.<br>2 bytes are used to protect a total of 64K bytes of main memory.<br>WRP0: Sectors 0-32K memory write protection control;<br>WRP1: Sectors 32-64K memory write protection control; | 0xFFFFFFFF |

### 19.5.1 User Option Bytes Unlock

The user option bytes operation can be unlocked by writing the sequence to the FLASH_OBKEYR register. After unlocking, the OBWRE bit in the FLASH_CTLR register will be set to 1, indicating that user option bytes can be erased and programmed. By setting the OBWRE bit in the FLASH_CTLR register, it will be cleared to 0 by software to lock again.

Unlocking sequence:

1) Write KEY1 = 0x45670123 to the FLASH_OBKEYR register;

2) Write KEY2 = 0xCDEF89AB to the FLASH_OBKEYR register.

*Note: The user needs to unlock the 2 layers: "LOCK" and "OBWRE" for word selection.*

### 19.5.2 User Option Bytes Programming

When programming the user option byte, FPEC only uses the low byte in the half-word, and automatically calculates the high byte (the high byte is the anti-code of the low byte), and then starts the programming operation, which will ensure that the byte in the user-selected word and its anti-code are always correct.

1) Check the LOCK bit of FLASH_CTLR register, if it is '1', you need to perform the operation of "Unlock Flash Memory".

2) Check the FLASH_CTLR register FLOCK bit, if it is '1', you need to perform the "Fast Programming Mode Unlock" operation.

3) Check the BSY bit of the FLASH_STATR register to make sure there is no other programming operation in progress.

4) Set the FTPG bit of FLASH_CTLR register to '1' to enable fast page programming mode.

5) Set the BUFRST bit of the FLASH_CTLR register to perform a clear internal 256 byte buffer operation.

6) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to become '1' to indicate the end of clearing, and clear the EOP bit to 0

7) Write 8 bytes of data to the specified address continuously (4 bytes/operation, with the address offset of 4 each time), then set the BUFLOAD bit in the FLASH_CTLR register and load it into the buffer.

8) Wait for BSY of FLASH_STATR register to be '0' and write the next data.

9) Repeat steps 7-8 for 2 times, and load all 16 bytes of data into the buffer area (the addresses of the two operations should be continuous).

10) Write the first address of the fast programming page to the FLASH_ADDR register.

11) Set the STRT bit of the FLASH_CTLR register to '1' to start fast page programming.
12) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to become '1' to indicate the completion of one fast page programming and clear the EOP bit to 0.
13) Query the FLASH_STATR register to see if there is an error or read the programmed address data checksum.
14) End programming and clear FTPG bit to 0.

*Note: When the "Read Protected" in the Modify Selection Word becomes "Unprotected", a whole chip erase of the main memory will be performed automatically. If a selection other than "Read Protected" is modified, a full erase operation will not occur.*

### 19.5.3 User Option Bytes Erase

Directly erase the entire 128-byte user-selected word area.
1) Check the LOCK bit in the FLASH_CTLR register, if it is 1, the "Unlock Flash" operation needs to be performed.
2) Check the BSY bit of the FLASH_STATR register to confirm that there is no programming operation in progress.
3) Check the OBWRE bit of FLASH_CTLR register, if it is 0, you need to perform the "User Option Byte Unlock" operation.
4) Set the OBER bit of FLASH_CTLR register to '1', then set the STRT bit of FLASH_CTLR register to '1' to enable user select word erase.
5) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erasure, and clear the EOP bit to 0
6) Read the erase address data checksum.
7) End to clear the OBER bit to 0.

### 19.5.4 Release Read Protection

Whether the flash is read protected or not is determined by the user option byte. Reading the FLASH_OBR register, when the RDPRT bit is '1' indicates that the flash is currently in a read protected state and the flash is operationally protected by a series of security guards in the read protected state. The process of unprotecting a read is as follows:
1) Erase the entire user option byte area, at which point the read protection field RDPR, at which point the read protection is still in effect.
2) Program the user option byte to write the correct RDPR code 0xA5 to unprotect the flash memory from read protection. (This step will first cause the system to automatically perform an entire erase operation on the flash memory)
3) Perform a power-on reset to reload the option byte (Including the new RDPR code), at which point the read protection is removed.

### 19.5.5 Release Write Protection

Whether the flash memory is write-protected or not is determined by the user select word. Read the FLASH_WPR register, each bit represents 4K bytes of flash space, when the bit is '1' it means non-write protected status, when it is '0' it means write protected. The process of unwrite-protecting is as follows:
1) Erase the entire user select word area.
2) Write the correct RDPR code 0xA5 to allow read access;
3) Perform a system reset, reload the selection byte (Including the new WRPR[1:0] byte) and the write protection is removed.

# Chapter 20 Extended Configuration (EXTEN)

## 20.1 Extended Configuration

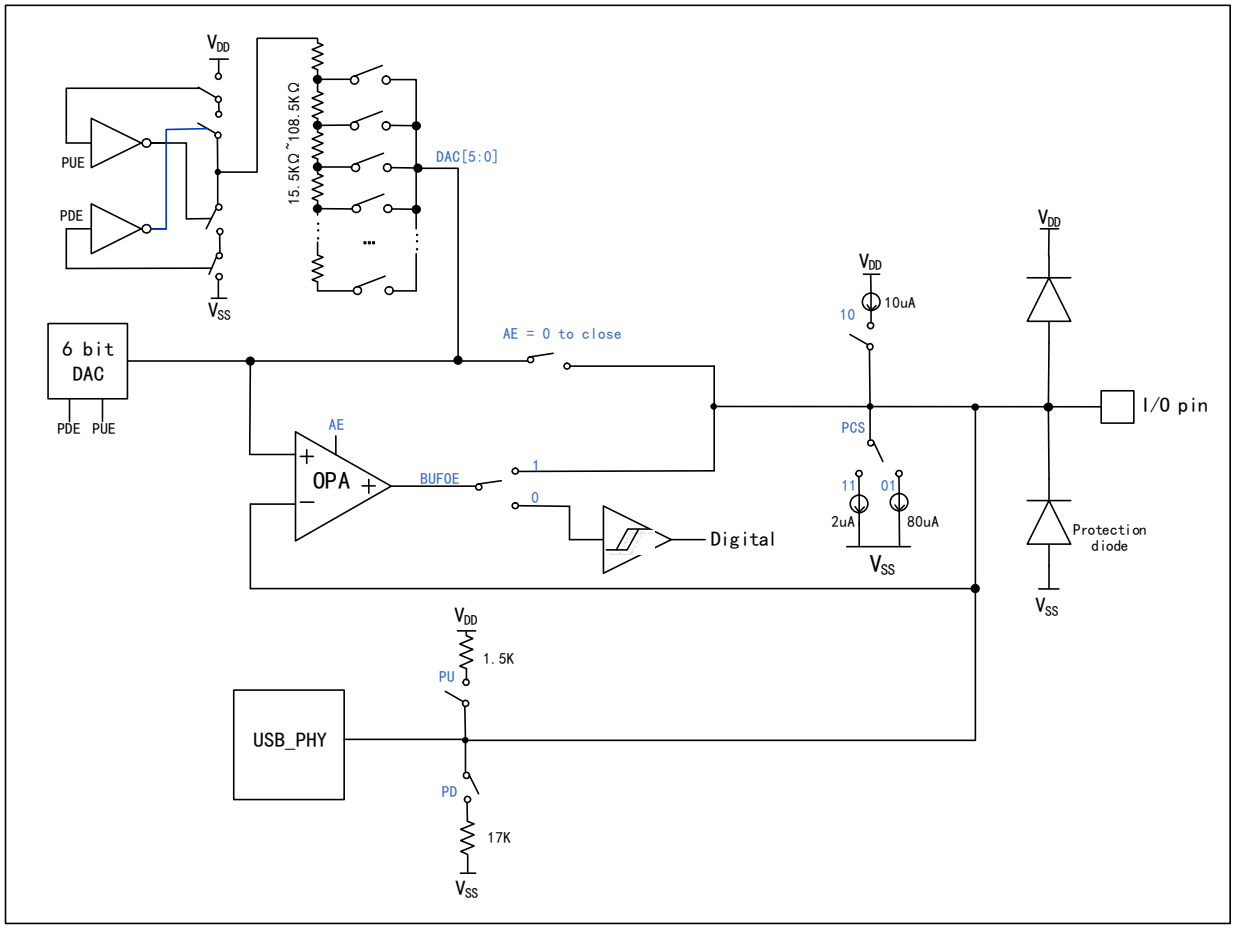
The system provides 3 EXTEN extended configuration units (EXTEN_CTLR0 register, EXTEN_CTLR1 register, EXTEN_CTLR2 register), which use HB clock and only perform reset actions when the system is reset.

EXTEN_CTLR0 register includes 2 source current control bits, 2 auxiliary control bits of USBPD and CC pins, and LOCKUP function monitoring function; Among them, the source current module can be used for external low-cost NTC temperature-sensing resistor, etc., and the temperature can be calculated by ADC; LOCKUP function monitoring function: when the LKUPEN field is enabled, the Lock-up situation monitoring of the system will be turned on. Once the Lock-up situation occurs, the system will reset the software and set the LKUPRST field to 1. After reading, you can write 1 to clear this flag.

EXTEN_CTLR1 register includes control bits of UDM/UDP pin of BC interface, DAC and DAC_BUF output buffer/comparator, and control bits of port pull-up and pull-down current; Each pin of BC interface contains a 6-bit DAC and an output buffer DAC_BUF. Among them, DAC can not only output programmable voltage, but also realize programmable pull-up resistance or pull-down resistance; DAC_BUF can not only buffer the DAC voltage and output it to the pin, but also be used as an analog voltage comparator to compare the pin voltage with the DAC voltage. The BC interface structure is shown in Figure 20-1.

The EXTEN_CTLR2 register includes 2 10-bit current selection bits with programmable sink current. The programmable current injection module ISINK supports 10-bit current precision, and can be used for high-precision voltage regulation of external DC-DC in 20mV steps to realize PPS protocol.

Figure 20-1 BC interface structure diagram




## 20.2 Functional Description

As shown in the BC interface structure diagram in Figure 20-1, the following modes can be configured through the UDM/UDP corresponding control bits of the EXTEN_CTLR1 register:

**DAC mode**

1. When PDE=1, PUE=1, AE=0, DAC[5:0] input data, BUFOE=0, this is the DAC mode without buffer.
2. When PDE=1, PUE=1, AE=1, DAC[5:0] input data, BUFOE=1, this is DAC mode with buffer.

The DAC mode output voltage DAC_OUT is:

$$DAC_OUT=DAC[5:0]*V_{DD33}/64$$

**CMP mode**

When PDE=1, PUE=1, AE=1, DAC[5:0] input data, BUFOE=0, this is the comparator mode, the input of the positive end of the comparator is DAC_OUT, the input voltage of the negative end of the comparator is the IO port, and the output signal of the comparator is sent to the digital terminal. if the IO input voltage is less than DAC_OUT then the comparator outputs 1, the IO input voltage is greater than DAC_OUT then the Comparator output 0.

**Pull-up/pull-down resistance mode**

1. When PDE=0, PUE=1, AE=0, DAC[5:0] input data, BUFOE=0, this is the programmable pull-up resistor mode. Where, when DAC[5:0]=000001, the pull-up resistor is equal to $7*R_{DAC}$ (about 108.5kΩ), and when

DAC[5:0]=111111, the pull-up resistor is equal to 1366/1365*$R_{DAC}$ (about 15.5kΩ).

2. When PDE=1, PUE=0, AE=0, DAC[5:0] input data, BUFOE=0, this is programmable pull-down resistor mode. Where, when DAC[5:0]=000000, the pull-down resistor is equal to $R_{DAC}$ (about 15.5kΩ), and when DAC[5:0]=111111, the pull-down resistor is equal to 7*$R_{DAC}$ (about 108.5kΩ).

*Note: See Table 20-1 for the resistance values of pull-up/pull-down resistors.*

**Pull-up/pull-down current mode**

1. When PCS = 10, this time for the weak pull-up current $I_{PU2}$, about 10uA.
2. When PCS = 11, this time for the weak pull-down current $I_{PD3}$, about 2uA.
3. When PCS=01, this is the weak pull-down current $I_{PD1}$, which is about 80uA.

Table 20-1 Lookup Table of Programmable Pull-up/Pull-down Resistors

| DAC[5:0] | Pull-up resistance (KΩ) | Pull-down resistance (KΩ) | DAC[5:0] | Pull-up resistance (KΩ) | Pull-down resistance (KΩ) |
|---|---|---|---|---|---|
| 000000 | Invalid | 1*$R_{DAC}$ | 100000 | 2*$R_{DAC}$ | 2*$R_{DAC}$ |
| 000001 | 7*$R_{DAC}$ | 1366/1365*$R_{DAC}$ | 100001 | 14/9*$R_{DAC}$ | 683/341*$R_{DAC}$ |
| 000010 | 6*$R_{DAC}$ | 342/341*$R_{DAC}$ | 100010 | 3/2*$R_{DAC}$ | 171/85*$R_{DAC}$ |
| 000011 | 26/5*$R_{DAC}$ | 214/213*$R_{DAC}$ | 100011 | 13/9*$R_{DAC}$ | 107/53*$R_{DAC}$ |
| 000100 | 5*$R_{DAC}$ | 86/85*$R_{DAC}$ | 100100 | 10/7*$R_{DAC}$ | 43/21*$R_{DAC}$ |
| 000101 | 13/3*$R_{DAC}$ | 226/223*$R_{DAC}$ | 100101 | 26/19*$R_{DAC}$ | 113/55*$R_{DAC}$ |
| 000110 | 21/5*$R_{DAC}$ | 54/53*$R_{DAC}$ | 100110 | 42/31*$R_{DAC}$ | 27/13*$R_{DAC}$ |
| 000111 | 85/21*$R_{DAC}$ | 130/127*$R_{DAC}$ | 100111 | 170/127*$R_{DAC}$ | 65/31*$R_{DAC}$ |
| 001000 | 4*$R_{DAC}$ | 22/21*$R_{DAC}$ | 101000 | 4/3*$R_{DAC}$ | 11/5*$R_{DAC}$ |
| 001001 | 24/7*$R_{DAC}$ | 234/223*$R_{DAC}$ | 101001 | 24/19*$R_{DAC}$ | 117/53*$R_{DAC}$ |
| 001010 | 10/3*$R_{DAC}$ | 58/55*$R_{DAC}$ | 101010 | 5/4*$R_{DAC}$ | 29/13*$R_{DAC}$ |
| 001011 | 42/13*$R_{DAC}$ | 18/17*$R_{DAC}$ | 101011 | 21/17*$R_{DAC}$ | 9/4*$R_{DAC}$ |
| 001100 | 16/5*$R_{DAC}$ | 14/13*$R_{DAC}$ | 101100 | 16/13*$R_{DAC}$ | 7/3*$R_{DAC}$ |
| 001101 | 40/13*$R_{DAC}$ | 146/135*$R_{DAC}$ | 101101 | 40/33*$R_{DAC}$ | 73/31*$R_{DAC}$ |
| 001110 | 64/21*$R_{DAC}$ | 34/31*$R_{DAC}$ | 101110 | 64/33*$R_{DAC}$ | 17/7*$R_{DAC}$ |
| 001111 | 256/85*$R_{DAC}$ | 10/9*$R_{DAC}$ | 101111 | 256/213*$R_{DAC}$ | 5/2*$R_{DAC}$ |
| 010000 | 3*$R_{DAC}$ | 6/5*$R_{DAC}$ | 110000 | 6/5*$R_{DAC}$ | 3*$R_{DAC}$ |
| 010001 | 5/2*$R_{DAC}$ | 256/213*$R_{DAC}$ | 110001 | 10/9*$R_{DAC}$ | 256/85*$R_{DAC}$ |
| 010010 | 17/7*$R_{DAC}$ | 64/33*$R_{DAC}$ | 110010 | 34/31*$R_{DAC}$ | 64/21*$R_{DAC}$ |
| 010011 | 73/31*$R_{DAC}$ | 40/33*$R_{DAC}$ | 110011 | 146/135*$R_{DAC}$ | 40/13*$R_{DAC}$ |
| 010100 | 7/3*$R_{DAC}$ | 16/13*$R_{DAC}$ | 110100 | 14/13*$R_{DAC}$ | 16/5*$R_{DAC}$ |
| 010101 | 9/4*$R_{DAC}$ | 21/17*$R_{DAC}$ | 110101 | 18/17*$R_{DAC}$ | 42/13*$R_{DAC}$ |
| 010110 | 29/13*$R_{DAC}$ | 5/4*$R_{DAC}$ | 110110 | 58/55*$R_{DAC}$ | 10/3*$R_{DAC}$ |
| 010111 | 117/53*$R_{DAC}$ | 24/19*$R_{DAC}$ | 110111 | 234/223*$R_{DAC}$ | 24/7*$R_{DAC}$ |
| 011000 | 11/5*$R_{DAC}$ | 4/3*$R_{DAC}$ | 111000 | 22/21*$R_{DAC}$ | 4*$R_{DAC}$ |
| 011001 | 65/31*$R_{DAC}$ | 170/127*$R_{DAC}$ | 111001 | 130/127*$R_{DAC}$ | 85/21*$R_{DAC}$ |
| 011010 | 27/13*$R_{DAC}$ | 42/31*$R_{DAC}$ | 111010 | 54/53*$R_{DAC}$ | 21/5*$R_{DAC}$ |
| 011011 | 113/55*$R_{DAC}$ | 26/19*$R_{DAC}$ | 111011 | 226/223*$R_{DAC}$ | 13/3*$R_{DAC}$ |
| 011100 | 43/21*$R_{DAC}$ | 10/7*$R_{DAC}$ | 111100 | 86/85*$R_{DAC}$ | 5*$R_{DAC}$ |
| 011101 | 107/53*$R_{DAC}$ | 13/9*$R_{DAC}$ | 111101 | 214/213*$R_{DAC}$ | 26/5*$R_{DAC}$ |
| 011110 | 171/85*$R_{DAC}$ | 3/2*$R_{DAC}$ | 111110 | 342/341*$R_{DAC}$ | 6*$R_{DAC}$ |
| 011111 | 683/341*$R_{DAC}$ | 14/9*$R_{DAC}$ | 111111 | 1366/1365*$R_{DAC}$ | 7*$R_{DAC}$ |

*Note: $R_{DAC}$ is the input impedance of DAC.*

## 20.3 Register Description

Table 20-2 EXTEN-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R32_EXTEN_CTLR0 | 0x40023800 | Configure extended control register 0 | 0x00A38040 |
| R32_EXTEN_KEYR | 0x40023804 | Configure unlock register | 0xXXXXXXXX |
| R32_EXTEN_CTLR1 | 0x40023808 | Configure extended control register 1 | 0xX000X000 |
| R32_EXTEN_CTLR2 | 0x4002380C | Configure extended control register 2 | 0x00XX00XX |
| R16_ISINK1_CFGR | 0x4002380C | ISINK1 configuration register | 0x00XX |
| R16_ISINK2_CFGR | 0x4002380E | ISINK2 configuration register | 0x00XX |
| R32_ISINK_ADJ | 0x1FFFF390 | ISINK irrigation current calibration value register | 0x00XX00XX |

### 20.3.1 Configure Extended Control Register 0 (EXTEN_CTLR0)

Offset address: 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | ISRC2_SEL | ISRC2_EN | ISRC1_SEL | ISRC1_EN | USBPD1_LVE_T[1:0] | | USBPD0_LVE_T[1:0] | | USBPD1_CC_HVT | USBPD0_CC_HVT | USBPD1_CC_REF | USBPD0_CC_REF |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WR_LOCK | Reserved | | | | | | | LKUPRST | LKUPEN | Reserved | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31:28 | Reserved | RO | Reserved | 0 |
| 27 | ISRC2_SEL | RW | ISOURCE2 source current selection bit:<br>1: 32uA;<br>0: 8uA. | 0 |
| 26 | ISRC2_EN | RW | ISOURCE2 source current enable bit (Need to configure ISINKEN=1):<br>1: On;;<br>0: Off. | 0 |
| 25 | ISRC1_SEL | RW | ISOURCE1 source current selection bit:<br>1: 32uA;<br>0: 8uA. | 0 |
| 24 | ISRC1_EN | RW | ISOURCE1 source current enable bit (Need to configure ISINKEN=1):<br>1: On;;<br>0: Off. | 0 |
| [23:20] | Reserved | RW | Reserved, the original value 1010b must be kept when writing. | 0xA |
| 19 | USBPD1_CC_HVT | RW | USBPD1 pin CC high threshold input mode enable:<br>1: High threshold input to reduce power consumption during PD1 communication.<br>0: Normal GPIO threshold input. | 0 |

| | | | | |
|---|---|---|---|---|
| 18 | USBPD0_CC_HVT | RW | USBPD0 pin CC high threshold input mode enable:<br>1: High threshold input to reduce power consumption during PD0 communication.<br>0: Normal GPIO threshold input. | 0 |
| 17 | USBPD1_CC_REF | RW | The CC pin of USBPD1 simulates a reference enable:<br>1: Reference enable, recommended to be normally open;<br>0: reference off. | 1 |
| 16 | USBPD0_CC_REF | RW | The CC pin of USBPD0 simulates a reference enable:<br>1: Reference enable, recommended to be normally open;<br>0: reference off. | 1 |
| 15 | WR_LOCK | RW1 | Control registers 0, 1 configure operation uplock:<br>1: Control registers 0, 1 configuration operation locked;<br>0: Control register 0, 1 configuration operation unlocked. | 1 |
| [14:8] | Reserved | RO | Reserved | 0 |
| 7 | LKUPRST | RW1 | LOCKUP reset flag:<br>1: LOCKUP occurred causing a system reset, write 1 to clear;<br>0: Normal. | 0 |
| 6 | LKUPEN | RW | LOCKUP monitoring function:<br>1: Enabled, performs a reset and sets LOCKUP_RST when a lock-up occurs in the system;<br>0: not enabled. | 1 |
| [5:0] | Reserved | RO | Reserved | 0 |

### 20.3.2 Configure Unlock Register (EXTEN_KEYR)

Offset address: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEYR[31:16] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEYR[15:0] | | | | | | | | | | | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | KEYR[31:0] | RW | Enter the following sequence of configuration operations to unlock controllers 0 and 1:<br>KEY1=0x45670123;<br>KEY2=0xCDEF89AB。 | X |

### 20.3.3 Configure Extended Control Register 1 (EXTEN_CTLR1)

Offset address: 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UDU | UD | UD | UD | UD | UDM_DAC[5:0] | | | | | | UD | UD | UDM_PC | | UD |

| _SHRT | M_IE | M_RAT | M_AI | M_AE | | | | | | | M_PDE | M_PUE | S[1:0] | | M_BUFOE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | UDP_IE | UDP_RAT | UDP_AI | UDP_AE | UDP_DAC[5:0] | | | | | | UDP_PDE | UDP_PUE | UDP_PCS[1:0] | | UDP_BUFOE |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 31 | UDU_SHRT | RW | UDP pin and UDM pin short enable bit:<br>1: Shorted;<br>0: No shorting. | 0 |
| 30 | UDM_IE | RW | Input enable when UDM pin is used as GPIO:<br>1: IO can be input;<br>0: IO input is always 0. | 1 |
| 29 | UDM_RAT | RW | UDM pin speed selection.<br>1: 20MHz;<br>0: 30MHz. | 0 |
| 28 | UDM_AI | RO | Output of the UDM pin comparator DAC_BUF. | x |
| 27 | UDM_AE | RW | UDM pin buffer/comparator DAC_BUF enable:<br>1: Output buffer/input comparator enable;<br>0: output buffer/input comparator off. | 0 |
| [26:21] | UDM_DAC[5:0] | RW | UDM pin 6-bit DAC data or control data from programmable pull-up or pull-down resistors. | 0 |
| 20 | UDM_PDE | RW | Mode selection for UDM port DAC and programmable pull-up and pull-down:<br>00: Disable DAC, disable programmable pull-up and pull-down resistors;<br>PDE=0, PUE=1: programmable pull-up resistor mode;<br>PDE=1, PUE=0: programmable pull-down resistor mode;<br>11: DAC programmable voltage mode for reference voltage of comparator or output to pin directly or output to pin via buffer, if UDM_AE=0 then output to pin directly, if UDM_AE=1 and UDM_BUFOE=1 then output to pin via buffer, if UDM_AE=1 and UDM_BUFOE=0 then compare pin voltage with DAC voltage Comparison is made and the result is UDM_AI. | 0 |
| 19 | UDM_PUE | RW | | 0 |
| [18:17] | UDM_PCS[1:0] | RW | Selects the pull-up or pull-down current built into the UDM pin:<br>00: No pull-up, no pull-down;<br>01: Rated pull-down 80uA;<br>10: Rated pull-up 10uA;<br>11: Rated pull-down 2uA. | 0 |
| 16 | UDM_BUFOE | RW | UDM pin DAC_BUF output enable:<br>1: UDM pin DAC_BUF output enable;<br>0: UDM pin DAC_BUF output off. | 0 |
| 15 | Reserved | RO | Reserved | 0 |
| 14 | UDP_IE | RW | U Input enable when DP pin is used as GPIO:<br>1: IO can be input;<br>0: IO input is always 0. | 1 |

| | | | | |
|---|---|---|---|---|
| 13 | UDP_RAT | RW | UDP pin speed selection:<br>1: 20MHz;<br>0: 30MHz. | 0 |
| 12 | UDP_AI | RO | Output of the UDP pin comparator DAC_BUF. | x |
| 11 | UDP_AE | RW | UDP pin buffer/comparator DAC_BUF enable:<br>1: Output buffer/input comparator enable;<br>0: output buffer/input comparator off. | 0 |
| [10:5] | UDP_DAC[5:0] | RW | UDP pin 6-bit DAC data or control data with programmable pull-up or pull-down resistors. | 0 |
| 4 | UDP_PDE | RW | Mode selection for UDP port DAC and programmable pull-up and pull-down:<br>00: Disable DAC, disable programmable pull-up and pull-down resistors;<br>PDE=0, PUE=1: programmable pull-up resistor mode;<br>PDE=1, PUE=0: programmable pull-down resistor mode;<br>11: DAC programmable voltage mode, used for the reference voltage of the comparator or output to the pin directly or output to the pin via buffer, if UDP_AE=0 then output to the pin directly, if UDP_AE=1 and UDP_BUFOE=1 then output to the pin via buffer, if UDP_AE=1 and UDP_BUFOE=0 then output the pin voltage with DAC voltage Comparison is made and the result is UDP_AI. | 0 |
| 3 | UDP_PUE | RW | | 0 |
| [2:1] | UDP_PCS[1:0] | RW | Selects the pull-up or pull-down current built into the UDP pin:<br>00: No pull-up, no pull-down;<br>01: Rated pull-down 80uA;<br>10: Rated pull-up 10uA;<br>11: Rated pull-down 2uA. | 0 |
| 0 | UDP_BUFOE | RW | UDP pin DAC_BUF output enable:<br>1: UDP pin DAC_BUF output enable;<br>0: UDP pin DAC_BUF output off. | 0 |

### 20.3.4 Configure Extended Control Register 2 (EXTEN_CTLR2)

Offset address: 0x0C

| Bit | Name |
|---|---|
| [31:16] | R16_ISINK2_CFGR |
| [15:0] | R16_ISINK1_CFGR |

### 20.3.5 ISINK1 Configuration Register (ISINK1_CFGR)

Offset address: 0x0C

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:6] | ISINK1_DAT[9:0] | WO | ISINK1 sink current programmable data bit: 0.25uA/bit。 | 0 |
| [5:0] | ISINK1_ADJ[5:0] | WO | Set the calibration value of ISINK1 sink current. | x |

*Note:*

*(1) ISINK1_CFGR is not readable, when rewriting the registers, the value of ISINK1_ADJ[5:0] comes from the corresponding bit in the ISINK Irrigation Current Calibration Value Register R32_ISINK_ADJ, and the value of ISINK1_DAT[9:0] is the new current data of ISINK1, for details, please refer to the EVT routine;*

*(2) The ISINKEN bit needs to be turned on when using the programmable irrigation current module.*

### 20.3.6 ISINK2 Configuration Register (ISINK2_CFGR)

Offset address: 0x0E

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:6] | ISINK2_DAT[9:0] | WO | ISINK2 sink current programmable data bit: 0.25uA/bit。 | 0 |
| [5:0] | ISINK2_ADJ[5:0] | WO | Set the calibration value of ISINK2 sink current. | x |

*Note:*

*(1) ISINK2_CFGR is not readable, when rewriting the registers, the value of ISINK2_ADJ[5:0] comes from the corresponding bit in the ISINK Irrigation Current Calibration Value Register R32_ISINK_ADJ, and the value of ISINK2_DAT[9:0] is the new current data of ISINK2, for details, please refer to the EVT routine;*

*(2) The ISINKEN bit needs to be turned on when using the programmable irrigation current module.*

### 20.3.7 ISINK Sink Current Calibration Value Register (ISINK_ADJ)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | ISINK2_ADJ[5:0] | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | ISINK1_ADJ[5:0] | | | | | |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:22] | Reserved | RO | Reserved | 0 |
| [21:16] | ISINK2_ADJ[5:0] | RO | ISINK2 sink current calibration value. | x |
| [15:6] | Reserved | RO | Reserved | 0 |
| [5:0] | ISINK1_ADJ[5:0] | RO | ISINK1 sink current calibration value. | x |

# Chapter 21 Debug Support

## 21.1 Main Features

This register allows the MCU to be configured in the debug state. includes

- Support counter of Window Watchdog (WWDG)
- Support counter of Timer

## 21.2 Register Description

### 21.2.1 Debug MCU Configuration Register 1 (DBGMCU_CR1)

Address: 0xE000D000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | TIM3_STOP | TIM2_STOP | TIM1_STOP | Reserved | | WWDG_STOP | Reserved |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:7] | Reserved | RW | Reserved | 0 |
| 6 | TIM3_STOP | RW | Timer 3 debug stop bit. The counter stops when the core enters the debug state.<br>1: Timer 3's counter stops working;<br>0: Timer 3's counter is still operating normally. | 0 |
| 5 | TIM2_STOP | RW | Timer 2 debug stop bit. The counter stops when the core enters the debug state.<br>1: Timer 2's counter stops working;<br>0: Timer 2's counter is still operating normally. | 0 |
| 4 | TIM1_STOP | RW | Timer 1 debug stop bit. The counter stops when the core enters the debug state.<br>1: Timer 1's counter stops working;<br>0: Timer 1's counter is still operating normally. | 0 |
| [3:2] | Reserved | RW | Reserved | 0 |
| 1 | WWDG_STOP | RW | Window watchdog debug stop bit. The debug window watchdog stops when the core enters the debug state.<br>1: The window watchdog counter stops working;<br>0: The window watchdog counter is still operating normally. | 0 |
| 0 | Reserved | RW | Reserved | 0 |

### 21.2.2 Debug MCU Configuration Register 2 (DBGMCU_CR2)

Address: 0xE000D004

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | STAND | STOP | SLEE |

| | | | |
|---|---|---|---|
| | BY | | P |

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [32:3] | Reserved | RW | Reserved | 0 |
| 2 | STANDBY | RW | Debugging standby mode bits.<br>1: (FCLK on, HCLK on) The digital circuitry section is not powered down and the FCLK and HCLK clocks are clocked by the internal RL oscillator. In addition, the microcontroller exits STANDBY mode by generating a system reset which is the same as a reset;<br>0: (FCLK off, HCLK off) The entire digital circuitry section is powered down.<br>From a software point of view, exiting STANDBY mode is the same as a reset (except that some status bits indicate that the microcontroller has just exited from the STANDBY state). | 0 |
| 1 | STOP | RW | Debug stop mode bits.<br>1: (FCLK on, HCLK on) In stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting stop mode, software must reconfigure the clock system to start the PLL, crystal, etc. (same operation as when configuring this bit to 0);<br>0: (FCLK off, HCLK off) When in STOP mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting from STOP mode, the clock configuration is the same as after a reset (the microcontroller is clocked by an 8MHz internal RC oscillator (HSI)). Therefore, software must reconfigure the clock control system to start the PLL, crystal, etc. | 0 |
| 0 | SLEEP | RW | Debug sleep mode bits.<br>1: (FCLK on, HCLK on) In sleep mode, both FCLK and HCLK clocks are supplied by the originally configured system clock;<br>0: (FCLK on, HCLK off) In sleep mode, FCLK is provided by the originally configured system clock and HCLK is off. Since sleep mode does not reset the configured clock system, the software does not need to reconfigure the clock system when exiting from sleep mode. | 0 |